Understanding the Test-Time Computing of Transformers: A Theoretical Study on In-Context Linear Regression

Anonymous Author(s)
Affiliation
Address
email

Abstract

Using more test-time computation during language model inference, such as generating more intermediate thoughts or sampling multiple candidate answers, has proven effective in significantly improving model performance. This paper takes an initial step toward bridging the gap between practical language model inference and theoretical transformer analysis by incorporating randomness and sampling. We focus on in-context linear regression with continuous/binary coefficients, where our framework simulates language model decoding through noise injection and binary coefficient sampling. Through this framework, we provide detailed analyses of widely adopted inference techniques. Supported by empirical results, our theoretical framework and analysis demonstrate the potential for offering new insights into understanding inference behaviors in real-world language models.

1 Introduction

Transformer-based [44] large language models (LLMs) have demonstrated impressive general-purpose capabilities, representing state-of-the-art architectures in natural language processing [14, 18, 1] and increasingly in other domains such as computer vision [37, 2]. While scaling laws for LLM training [24] have described their performance with respect to the train-time compute (i.e. model size, data size, and training time, e.g.), leveraging additional test-time computation of the pretrained LLMs, such as extend reasoning length by generating additional intermediate thoughts [50, 18, 36] or sampling multiple candidate answers and aggregating to obtain the best one [12, 49], has recently demonstrated great potential for further enhancing their reasoning capabilities. However, despite the success of scaling up test-time computing for LLMs, the theoretical understanding of transformer models, even for the relatively simpler linear cases, for such successes remains quite limited.

Due to the success of LLMs itself, a huge body of recent theory works has emerged, aiming at understanding the hidden mechanisms of transformers from other angles. These works have been focused on seeking to explain the model's capabilities in memorization [33, 25], in-context learning (ICL) [47, 57, 22], function approximation power [42, 34], algorithm simulation [10, 15, 31], and the training dynamics [53, 57, 9] for transformers initialized from scratch, to name a few. Most of these works consider simplified settings with linear attention [47] and focus on how transformers can *directly* leverage their output activations to solve specific tasks like in-context linear regression [16], ignoring the sampling and tokenization procedure for LM decoding, creating substantial gaps between theoretical analysis and practical LLM applications.

One of the main gap between prior theoretical works and LLM used in practice is that, prior theoretical works typically focus on transformers with deterministic decoding procedures, where the model output is fixed for a given prompt. In practice, many inference techniques for scaling up test-time

Submitted to 39th Conference on Neural Information Processing Systems (NeurIPS 2025). Do not distribute.

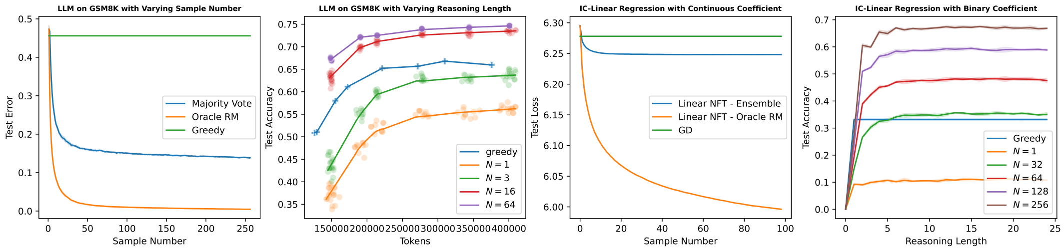

Figure 1: Comparison between real-world LLM's inference (above) and our designed sampling framework (below) for different sample numbers N and reasoning lengths. Our framework simulates language model decoding through noise injection and binary coefficient sampling, exhibiting trends similar to real-world LLMs' inference, Details can be found in Appendix B

computing, such as majority voting [49], best-of-N sampling (BoN) [12], and tree of thoughts (ToT) [54], rely on probabilistic sampling procedures in real-world LLMs: given a prompt, the model predicts subsequent tokens by first computing a distribution over potential candidates and then sampling from it. This gap between the theoretical setups and the real-world LLM behavior hinders us towards understanding and analyzing of the success of transformer test-time computation.

Our contributions. In this work, we aim to bridge the gap between practical language model (probabilistic) inference and theoretical transformer analysis, providing initial theoretical insights into transformer test-time computation. Specifically, we examine the in-context linear regression task with continuous/binary coefficients, simulate LLMs' sampling decoding procedure by injecting random noise (continuous case) or conducting discrete sampling (binary case) based on the model's original output, using the processed tokens for subsequent sampling decoding steps. We then conduct analysis towards test-time computation of transformers based on our theoretical framework. The main contributions of this paper are highlighted as follows:

- We take an initial step toward bridging the gap between practical language model inference and theoretical transformer analysis by incorporating randomness and sampling. Our framework simulates language model decoding through noise injection and binary coefficient sampling, exhibiting trends similar to real-world LLMs' inference, as demonstrated in Fig 1.
- Through our framework, we conduct detailed analysis of how test-time computation plays a role in our reasoning framework, including reasoning steps and sampling number, which can be applied to widely adopted inference techniques such as majority voting, ensembling, and chain-of-thought prompting.
- We validate our theoretical analysis through extensive experiments. Furthermore, we attempt to predict real-world LLM performance using our theoretical framework. The results demonstrate the potential of applying our theoretical framework for practical LLM behavior analysis.

Related Works. Our work is related to recent works on *scaling test-time computing in LLMs*, *theory for transformer test-time computing*, and *theory for transformer in-context learning*. Due to space limit, we defer them to Appendix A.

1.1 Preliminaries and More Backgrounds

This section outlines the problem setups. We first detail transformers' inference mechanism, emphasizing *sampling-based* techniques for enhancing test-time computation. We then introduce in-context linear regression, the theoretical task central to our study.

1.1.1 Transformer and Sampling-based Test-time Computing

A transformer [44] is an auto-regressive sequence-to-sequence model that predicts the next token's distribution, i.e., $p(x_{t+1}|x_t, \cdots, x_1)$. It maps the representation of the last token x_{t+1} to a softmax distribution over the vocabulary space $\mathcal{V}$ to determine the probability of x_{t+1}.

The above inference mechanism can be abstracted in the following way. Given the current input sequence embedding $\mathbf{H}_t = (\mathbf{h}_1, \cdots, \mathbf{h}_t) \in \mathbb{R}^{d_e \times t}$, one *iteratively* performs the following two steps:

- Compute and extract the hidden state for the last position t, i.e., $\widetilde{\mathbf{h}}_t = \mathtt{TF}_\theta(\mathbf{H}_t)$, where $\mathtt{TF}_\theta$ denotes the stacked transformer blocks in the whole architecture.
- Sample the next token x_{t+1} (and thus the embedding of the next token $\mathbf{h}_{t+1}$) based on a probability distribution returned by a sampling algorithm inputted with $\widetilde{\mathbf{h}}_t$, i.e., $\mathbf{h}_{t+1} \leftarrow \mathtt{Sampling\_Alg}(\widetilde{\mathbf{h}}_t)$.

Sampling-based test-time computation. As previously introduced, the probabilistic nature of the computation procedure can introduce randomness into the inference process, which is key to an array of techniques for scaling up test-time computing in order to boost the performance of large language models for various tasks, including Best-of-N sampling (BoN) [41, 35, 13], majority vote [48], etc. Notably, these methods typically sample N independent reasoning trajectories through the above decoding mechanism and choose the one with the highest value of a given reward model or the most consistent one across all candidates.

1.1.2 Theoretical Task: In-context Linear Regression.

We explore how sampling-based test-time computing can enhance transformer performance by focusing on *in-context linear regression*, a common problem setup [5, 46, 57, 11]. In-context learning (ICL [8]) involves auto-regressive models inferring answers from few task demonstrations. we consider the following general setup: first drawing the ground truth parameter from the prior $\mathbf{w}^* \sim p_{\mathbf{w}}(\cdot)$, then

$$(\mathbf{x}_i, y_i) \sim \mathbb{D}_{\mathbf{w}^*}, y_i = \mathbf{x}_i^\top \mathbf{w}^* + \epsilon_i, \ \forall i \in [n], \tag{1.1}$$

where $p_{\mathbf{w}}$ denotes the prior distribution of the regression tasks, ϵ_i is the i.i.d. random noise, and $n \in \mathbb{N}$ is the size of the in-context dataset. The goal of in-context linear regression is to use transformers to make predictions regarding the true label $\mathbf{x}_{\text{query}}^\top \mathbf{w}^*$ associated with another covariate $\mathbf{x}_{\text{query}} \sim \mathcal{N}(0, \mathbf{I}_d)$ when prompted with the in-context dataset $(\mathbf{x}_1, y_1, \cdots, \mathbf{x}_n, y_n)$ concatenated with the query $\mathbf{x}_{\text{query}}$. Towards such a goal, this work aims to establish a theoretical framework that allows one to principally investigate how sampling-based techniques for scaling up test-time computing could benefit the predictions, thus boosting the performance of solving the task.

2 Scaling Test-time Computation for In-Context Regression

In this section, we introduce our theoretical framework for studying sampling-based test-time computing of transformers (Section 1.1.1) through in-context linear regression (Section 1.1.2). We present our framework in Section 2.1. After that, we study two instances of the in-context linear regression task (1.1), depending on the types of the task prior $p_{\mathbf{w}}$, to design concrete sampling algorithms for inference.

2.1 A Theoretical Framework

We begin by noticing that most of the existing prior works on in-context linear regression by transformers are *incapable* for studying sampling-based test-time computing due to the lack of (i) randomness of the output of the transformer architecture they study; (ii) chain-of-thought (CoT) style multi-step reasoning in the outputs. To handle the challenge, we explicitly construct an inference mechanism that involves both randomness and auto-regressive CoT reasoning to solve in-context linear regression tasks. Specifically, motivated by the recent work of [22], we consider the specific goal of *in-context coefficient prediction*, where the final output of the transformer reasoning path is a prediction $\widehat{\mathbf{w}}$ of the task coefficient $\mathbf{w}^*$. The transformer inference mechanism is designed to output stochastic reasoning paths, and different sampling-based test-time computing techniques correspond to how to aggregate different reasoning paths.

Inputs and transformer architecture. Given the in-context dataset $(\mathbf{x}_1, y_1, \cdots, \mathbf{x}_n, y_n)$, the prompt to the transformer (defined later) is the following matrix in $\mathbb{R}^{d_e \times (n+1)}$,

$$\mathbf{H}_0 = \begin{pmatrix} \mathbf{x}_1 & \cdots & \mathbf{x}_n & \mathbf{0} \\ y_1 & \cdots & y_n & 0 \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{w}_0 \\ 0 & \cdots & 0 & 1 \end{pmatrix} := \begin{pmatrix} \mathbf{X}^\top & \mathbf{0} \\ \mathbf{y}^\top & 0 \\ \mathbf{0} & \mathbf{w}_0 \\ \mathbf{0} & 1 \end{pmatrix}, \tag{2.1}$$

where the dimension of the embedding $d_e = 2d + 2$. We denote $\mathbf{X}^\top = (\mathbf{x}_1, \cdots, \mathbf{x}_n) \in \mathbb{R}^{d \times n}$ as the collection of covariates, and denote $\mathbf{y}^\top = (y_1, \cdots, y_n) \in \mathbb{R}^{1 \times n}$ as the collection of labels. We input an initial guess of the coefficient, denoted by $\mathbf{w}_0$, and we $\mathbf{w}_0 = \mathbf{0}$ without loss of generality. Note that such a prompt embedding format which separates the space of data and the space of weight predictions follows the convention of [6, 22] in order to facilitate theoretical analysis.

The model we consider is a one-layer self-attention module equipped with residual connection [46, 57, 4, 22]:

$$\mathtt{TF}_\theta(\mathbf{H}) := \mathbf{H} + \mathbf{V}\mathbf{H} \cdot \frac{\mathbf{H}^\top \mathbf{W}\mathbf{H}}{n} : \mathbb{R}^{d_e \times *} \mapsto \mathbb{R}^{d_e \times *}. \tag{2.2}$$

where $\theta = \{\mathbf{V}, \mathbf{W}\}$ denotes the parameters. Here $\mathbf{V} \in \mathbb{R}^{d_e \times d_e}$ represents the consolidation of the projection and value matrices in a standard transformer block, and $\mathbf{W} \in \mathbb{R}^{d_e \times d_e}$ denotes the consolidation of the key and query matrices.

Sampling-based auto-regressive inference mechanism. With the model (2.2) and the prompt (2.1), we consider the following mechanism of inference that mimics a real LLM.

Definition 2.1 (Inference mechanism)**.** *Given a prompt embedding matrix $\mathbf{H}_0$, for each $\ell \in \mathbb{N}$, we iteratively sample the embeddings for the next token as following:*

- *Compute $\widetilde{\mathbf{H}}_\ell = \mathtt{TF}_\theta(\mathbf{H}_\ell)$ with $\mathtt{TF}_\theta(\mathbf{H}_\ell)$ defined in* (2.2)*;*
- *Extract $\widetilde{\mathbf{h}}_\ell$ from $\widetilde{\mathbf{H}}_\ell$ last column, i.e., $\widetilde{\mathbf{h}}_\ell = (\widetilde{\mathbf{H}}_\ell)_{:,-1}$;*
- *Sample the embedding vector for the next token via* `Sampling_Alg`*, i.e., $\mathbf{h}_{\ell+1} \leftarrow$* `Sampling_Alg`$(\widetilde{\mathbf{h}}_\ell)$*;*
- *Concatenate to obtain the embedding matrix for the new sequence of length $\ell+1$, i.e., $\mathbf{H}_{\ell+1} = (\mathbf{H}_\ell, \mathbf{h}_{\ell+1})$.*

Here `Sampling_Alg`$(\cdot)$ is to be determined that assigns the distribution of the next token (embedding) conditioning on the last token's embedding output by the transformer. Note that the output of the above mechanism is a joint result of the transformer model and the sampling algorithm.

Towards the goal of in-context weight prediction for (1.1), we introduce the following proposition, which shows that the transformer architecture together with a proper sampling algorithm can implement variants of *noisy gradient descent*.

Proposition 2.2 (Definition 2.1 can implement noisy GD)**.** *There exists a transformer instance of* (2.2) *denoted by $\mathtt{TF}_{\theta_{\mathtt{GD}}}$ and a type of sampling algorithm* `Sampling_Alg` *such that given prompt $\mathbf{H}_0$ defined in* (2.1)*, the output embedding after t iterative generations $\mathbf{H}_t$ according to Definition 2.1 satisfies $(\mathbf{H}_t)_{:,n+\ell} = (\mathbf{0}^\top, 0, \mathbf{w}_\ell^\top, 1)^\top$ with*

$$\mathbf{w}_\ell \sim p\left(\cdot \middle| \mathbf{w}_{\ell-1} - \frac{\eta}{n} \cdot \mathbf{X}^\top \left(\mathbf{X}\mathbf{w}_{\ell-1} - \mathbf{y}\right)\right), \forall 1 \le \ell \le t, \tag{2.3}$$

where the conditional distribution $p(\cdot|\cdot)$ is specified by the sampling algorithm `Sampling_Alg`.

This proposition is mainly motivated by the recent work of [22]. Please refer to Appendix C.1 for a detailed proof of Proposition 2.2. Proposition 2.2 shows that the above inference mechanism is able to explicitly implement gradient-based iterative algorithms to predict the regression coefficient $\mathbf{w}^*$. We define the prediction of the regression coefficient after t reasoning steps of one reasoning path as $\mathbf{w}_t := (\mathbf{H}_t)_{d+2:2d+1,n+t}$. One special case of Proposition 2.2 is a transformer that explicitly performs standard multi-step GD [22], i.e., $p(\cdot|x) = \delta_x(\cdot)$. Please see Appendix C.2 for the details.

Now to theoretically understand the effectiveness of more sophisticated sampling-based test-time computing techniques, e.g., Best-of-N and majority vote, we go beyond (C.6) and consider sampling algorithms that does introduce randomness into the reasoning path. We formalize these test-time computing methods we study in this paper as following.

Definition 2.3 (Sampling-based test-time computing techniques)**.** *Given a transformer $\mathtt{TF}_\theta$ and a sampling algorithm that jointly satisfy Proposition 2.2, together with a prompt embedding matrix $\mathbf{H}_0$ in* (2.1)*, a CoT reasoning length limit $t \in \mathbb{N}_+$, and a sampling budget $N \in \mathbb{N}_+$, we consider the following test-time computing methods:*

- *Firstly generate N random predictions of the regression coefficient as $\{\mathbf{w}_t^{(j)}\}_{j=1}^N$ (see Proposition 2.2);*
- *Then aggregate the N random outcomes $\{\mathbf{w}_t^{(j)}\}_{j=1}^N$ by using one of the following options:*
  1. *Ensemble: $\mathbf{w}_{\mathtt{avg}} := N^{-1} \cdot \sum_{j=1}^N \mathbf{w}_t^{(j)}$;*
  2. *Best-of-N: $\mathbf{w}_{\mathtt{BoN}} := \arg\max_{\{\mathbf{w}_t^{(j)}\}_{j=1}^N} R(\mathbf{w}_t^{(j)})$ where $R(\cdot) : \mathbb{R}^d \mapsto \mathbb{R}$ is certain reward function;*
  3. *Majority vote: $\mathbf{w}_{\mathtt{mv}} := \arg\max_{\{\mathbf{w}_t^{(j)}\}_{j=1}^N} \mathtt{Occur}(\mathbf{w}_t^{(j)})$, where $\mathtt{Occur}(\cdot) : \mathbb{R}^d \mapsto \mathbb{N}$ is a proper function that counts the occurrence of the input.*

In the following Sections 2.2 and 2.3, we instantiate the in-context linear regression task (1.1) to more concrete task priors, and investigate the effectiveness and the scaling law of the above test-time computing techniques. We also remark that in this paper we assume the existence of a transformer satisfying Proposition 2.2 without explicitly training such one from scratch, which is left as an interesting future work.

2.2 Case Study 1: In-context Linear Regression with Continuous Coefficient

The first type of tasks we consider is the standard in-context linear regression with continuous regression coefficient sampled from a Gaussian distribution, i.e., $p_{\mathbf{w}} = \mathcal{N}(\mathbf{0}, \omega^2 \cdot \mathbf{I}_d)$. For this case, the specific type of sampling algorithms `Sampling_Alg` we study is concluded in Algorithm 1.

Algorithm 1 Sampling algorithm for in-context linear regression with continuous coefficient

1: **Input:** token embedding $\widetilde{\mathbf{h}}$, noise level $\sigma \geq 0$, noise transformation function $\phi_{\cdot}(\cdot) : \mathbb{R}^d \times \mathbb{R}^d \mapsto \mathbb{R}^d$.
2: Extract the coefficient $\widetilde{\mathbf{w}}$ from $\widetilde{\mathbf{h}}$, i.e., $\widetilde{\mathbf{w}} = (\widetilde{\mathbf{h}})_{d+2:2d+1}$
3: Sample a noise vector $\boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}, \sigma^2 \cdot \mathbf{I}_d)$
4: Define $\mathbf{w} \leftarrow \widetilde{\mathbf{w}} + \phi_{\boldsymbol{\xi}}(\widetilde{\mathbf{w}})$
5: **Output:** $\mathbf{h} := (\mathbf{0}, 0, \mathbf{w}, 1)^\top$.

Under sampling method Algorithm 1, Proposition 2.2 is satisfied with $x' \sim p(\cdot|x)$ given by $x' = x + \phi_{\boldsymbol{\xi}}(x)$ for a Gaussian random seed $\boldsymbol{\xi}$ and some noise transformation function $\phi_{\boldsymbol{\xi}}$. Recall that by Proposition 2.2, $\widetilde{\mathbf{w}}$ output by the transformer is performing one-step gradient descent from the last prediction. The intuition of studying Algorithm 1 is that such a noisy version of the gradient descent could allow exploration of the loss landscape, and we aim to investigate whether the test-time computing techniques in Definition 2.3 could properly aggregate the random gradient-based paths to achieve a better prediction than vanilla multi-step GD (C.6) via less overfitting. In this paper, we investigate the following two concrete and simple examples of the noise transformation function (NFT) $\phi_{\boldsymbol{\xi}}$. Potential future works could investigate other types of $\phi_{\boldsymbol{\xi}}$.

Example 2.4 (Constant NFT). $\phi_{\boldsymbol{\xi}}(\mathbf{w}) := \boldsymbol{\xi}$*, independent of the input* $\mathbf{w}$ *and is homogeneous across reasoning steps.*

Example 2.5 (Linear NFT). $\phi_{\boldsymbol{\xi}}(\mathbf{w}) := \boldsymbol{\xi}\boldsymbol{\xi}^\top \mathbf{w}$*, linear in the input predicted weight* $\mathbf{w}$ *such that the sampling distribution has different shape based upon the current decoding result.*

We consider the following test-time computing methods.

Baseline: multi-step GD with CoT (C.6)**.** This is a transformer implementing a vanilla GD, without using Algorithm 1 but directly using one-step GD as the next token. It is clear that this baseline is deterministic and does not require multiple samples.

Ensemble. We consider sample average of the predictions from N reasoning paths. We denote the resulting prediction after N sampling paths of length t as $\mathbf{w}_{\mathtt{avg}}$.

Best-of-N. We also consider BoN with the oracle reward model $R^\star(\mathbf{w}) := -\|\mathbf{w} - \mathbf{w}^*\|_2^2$. The resulting prediction accuracy gives an upper bound for other test-time computing method due to the usage of the truth. We denote the resulting prediction after N sampling paths of length t by $\mathbf{w}_{\mathtt{BoN}}$.

2.3 Case Study 2: In-context Sparse Linear Regression in Discrete Space

Motivated by the practical setting where the candidate tokens lie in a discrete space, we also consider another case in which the coefficient is a sparse binary vector, denoted as $\mathbf{w}^* \in \{0,1\}^d$ with $\|\mathbf{w}^*\|_0 = k < d$. In this situation, we consider the following sampling algorithm `Sampling_Alg`, which performs sampling on a discrete space $\{0,1\}^d$ based on the predicted weight $\widetilde{\mathbf{w}}$ in the transformer output. In algorithm 2, the function $\mathtt{ClipNorm}(\cdot)$ first clips each element in $\widetilde{\mathbf{w}}$ to be non-negative and then normalizes the resulting vector such that its elements sum to 1, i.e., $(\mathtt{ClipNorm}(\widetilde{\mathbf{w}}))_i = \max\{\widetilde{w}_i, 0\} / \sum_{i'=1}^d \max\{\widetilde{w}_{i'}, 0\}$. This resembles the softmax operation over a vocabulary set. Then algorithm 2 simulates LLM decoding by sampling tokens based such a distribution. More specifically, given the distribution p, we sample the (embedded) next token $\mathbf{w}$ as a k-sparse vector with non-zero coordinates sampled from p. We treat the vector sparsity k as a fixed parameter satisfying $1 \leq k < d$, with k typically set to 1 in practice. Such a discrete nature of these coefficients enables us to consider the method of majority vote among the sampling-based test-time computing strategies in Definition 2.3. In this work, we compare majority vote to a baseline inference mechanism based on greedy decoding which does not utilize sampling.

Algorithm 2 Sampling algorithm for in-context linear regression with binary coefficient

1: **Input:** token embedding $\widetilde{\mathbf{h}}$, coefficient sparsity $k \in [d]$.
2: Initialize $\mathbf{w} \leftarrow \mathbf{0}_d$
3: Extract the coefficient $\widetilde{\mathbf{w}}$ from $\widetilde{\mathbf{h}}$, i.e., $\widetilde{\mathbf{w}} = (\widetilde{\mathbf{h}})_{d+2:2d+1}$
4: Compute predicted distribution $p = \texttt{ClipNorm}(\widetilde{\mathbf{w}})$
5: Sample k different indices $(e_1, \ldots, e_k) \subset [d]$ based on p without replacement
6: Assign $w_{e_\ell} = 1$ for each $e_\ell \in \{e_1, \cdots, e_k\}$
7: **Output:** $\mathbf{h} := (\mathbf{0}, 0, \mathbf{w}, 1)^\top$.

Baseline: greedy decoding. In the decoding step, instead of sampling k items based on p as depicted in Algorithm 2 (Line 5), we opt to choose k items with the highest k probabilities under p and set the corresponding indices of $\mathbf{w}$ to 1. This mirrors the greedy decoding algorithm commonly used in practice. We denote the resulting prediction after t reasoning steps as $\mathbf{w}_t^{\texttt{greedy}}$.

Majority vote. Utilizing the discrete nature of the coefficients, we apply the $\texttt{Occur}(\cdot)$ function to candidate answers, selecting the most frequent one as our majority vote (see Definition 2.3). The prediction after sampling N reasoning paths of length t is denoted as $\mathbf{w}t, N^{\texttt{mv}}$.

Here we present theoretical results for Case Study 1 and 2 in Section 3 and 4 respectively, with numerical results in Section 5.1.

3 Analysis of In-context Linear Regression with Continuous Coefficient

In this section, we establish the theoretical analysis for Section 2.2. We measure the performance of any in-context coefficient prediction by its population risk under $\mathbb{D}_{\mathbf{w}^*}$, i.e., $L_{\mathbb{D}_{\mathbf{w}^*}}(\mathbf{w}) := (1/2) \cdot \mathbb{E}_{(\mathbf{x},y)\sim\mathbb{D}_{\mathbf{w}^*}}[(y - \mathbf{x}^\top\mathbf{w})^2]$, which is equivalent to consider the following excess risk,

$$\mathcal{E}(\mathbf{w}) := L_{\mathbb{D}_{\mathbf{w}^*}}(\mathbf{w}) - \inf_{\mathbf{w}' \in \mathbb{R}^d} L_{\mathbb{D}_{\mathbf{w}^*}}(\mathbf{w}) = \frac{1}{2} \cdot \|\mathbf{w} - \mathbf{w}^*\|_{\mathbf{H}}^2, \tag{3.1}$$

where $\mathbf{H} := \mathbb{E}_{\mathbf{x}\sim\mathbb{D}_{\mathbf{w}^*}}\left[\mathbf{x}\mathbf{x}^\top\right]$ denotes the population covariance matrix. We denote the collection of label noise in the in-context data as $\boldsymbol{\epsilon} := \mathbf{y} - \mathbf{X}\mathbf{w}^*$. We also denote the eigenvalues of the population covariance matrix $\mathbf{H}$ as $\{\lambda_i\}_{1\le i\le d}$ in a non-increasing order. Our analysis relies on standard assumptions on the data distribution [7], which is presented in Assumption E.1 due to space limit. By the same reason, we present our results for a special case of $\mathbf{H}$ with polynomially decaying eigenvalues, and refer to the readers to the expressions of general $\mathbf{H}$ in Appendix D.

Baseline: multi-step GD with CoT. The following result gives the excess risk bound for transformers implementing vanilla multi-step gradient descent (C.6). This is a corollary of Theorem D.1 and is proved in Appendix E.2.

Proposition 3.1. *Under the same assumptions and setups as in Theorem D.1, by additionally assuming that the spectrum of* $\mathbf{H}$ *satisfies polynomially decaying, i.e.,* $\lambda_i = i^{-(r+1)}$ *for some* $r \ge 1$*, then for any reasoning path length* $t \lesssim \eta(r+1)^{(r+1)/2} d^{(r+1)/2}$*, with probability at least* $1 - 1/\mathrm{poly}(n)$*,*

$$\mathbb{E}_{\boldsymbol{\epsilon},\mathbf{w}^*}\left[\mathcal{E}(\mathbf{w}_{\texttt{GD}})\right] \lesssim \omega^2 \cdot \left(\frac{1}{t\eta}\right)^{\frac{r}{r+1}} + \frac{\sigma_\epsilon^2}{n} \cdot (t\eta)^{\frac{1}{r+1}}. \tag{3.2}$$

Aggregating by ensembling. In this case, the final regression coefficient reasoned by the transformer test-time computing under the budget of CoT length t and reasoning path number N is explicitly given by $\mathbf{w}_{\texttt{avg}} := N^{-1} \cdot \sum_{j=1}^N \mathbf{w}_t^{(j)}$, where each random reasoning path $\{\mathbf{w}_\ell^{(j)}\}_{1\le\ell\le t}$ is i.i.d. generated according to Definition 2.1 via a transformer satisfying Proposition 2.2 and with Algorithm 1. The following result gives the excess risk bound for this method with different choices of the NFT $\phi_{\boldsymbol{\xi}}$. The proof is in Appendix E.4.

Theorem 3.2. *Under the same assumptions and setups as in Theorem D.2, additionally assuming that the spectrum of* $\mathbf{H}$ *satisfies polynomially decaying, i.e.,* $\lambda_i = i^{-(r+1)}$ *for some constant* $r \ge 0$*, we have the following results.*

1. *Constant noise transformation function (Example 2.4): taking the reasoning length* $t \lesssim \eta(r+1)^{(r+2)/2} n^{(r+1)/2}$*, with probability at least* $1 - 1/\mathrm{poly}(n)$*,*

$$\mathbb{E}\left[\mathcal{E}(\mathbf{w}_{\mathtt{avg}})\right] \lesssim \omega^2 \cdot \left(\frac{1}{t\eta}\right)^{\frac{r}{r+1}} + \frac{\sigma_\epsilon^2}{n} \cdot (t\eta)^{\frac{1}{r+1}} + \frac{\vartheta_{n,t}}{N}. \tag{3.3}$$

2. *Linear noise transformation function (Example 2.5): taking the noise variance* $\sigma^2 \asymp d^{-1}$*, the reasoning length* $t > \sigma^{-2} \cdot \log 2$*, with probability at least* $1 - 1/\mathrm{poly}(n)$*,*

$$\mathbb{E}\big[\mathcal{E}(\mathbf{w}_{\mathtt{avg}})\big] \lesssim \omega^2 \cdot \widetilde{\lambda}^{\frac{r}{r+1}} + \frac{\sigma_\epsilon^2}{n} \cdot \left(\frac{\eta(1-\sigma^2)}{\sigma^2}\right)^{\frac{1}{r+1}} + \frac{\varsigma_n}{N}, \tag{3.4}$$

where $\widetilde{\lambda} := \eta^{-1}(2t^{-1} + \sigma^2(1+2t^{-1})/(1-\sigma^2))$.

Here the expectation is taken with respect to $\boldsymbol{\epsilon}$*,* $\mathbf{w}^*$*, and all the sampling noise* $\boldsymbol{\xi}$ *across different reasoning steps and paths. The explicit formula for the functions* $\vartheta_{n,t}$ *and* ς_n *are deferred to* (D.4) *and* (D.8)*, respectively.*

The above theorem reveals how the prediction accuracy evolves as the reasoning length t and sample numbers N increase. In particular, we make the following remarks (i) In the above excess risk, the terms $\vartheta_{n,t}/N$ and ς_n/N represent the error from sampling finitely many reasoning paths N. By taking N large enough (see (D.9) and (D.12) in Corollary D.3), the leading term of the excess risk would be the first two terms. (ii) By the result for Example 2.4, Algorithm 1 with constant noise does not provide benefit compared with TF implementing vanilla GD (see Proposition 3.1). (iii) In contrast, we next show that with linear NFT Algorithm 1 can prevent overfitting to noisy labels. Considering the following regime of the parameters,

$$\omega, \sigma_\epsilon \asymp 1, \quad n \asymp \eta d, \quad \sigma^2 \asymp d^{-1}, \quad t \asymp \widetilde{t} \cdot \sigma^{-2}, \tag{3.5}$$

risk bounds for the vanilla multi-step GD and the ensemble method (using linear NFT (Example 2.5)) are as following,

$$\mathbb{E}_{\boldsymbol{\epsilon},\mathbf{w}^*}\left[\mathcal{E}(\mathbf{w}_{\mathtt{GD}})\right] \lesssim \widetilde{t}^{\frac{1}{r+1}} \cdot (\eta d)^{-\frac{r}{r+1}}, \tag{3.6}$$

$$\mathbb{E}_{\boldsymbol{\epsilon},\mathbf{w}^*,\boldsymbol{\xi}}\left[\mathcal{E}(\mathbf{w}_{\mathtt{avg}})\right] \lesssim (\eta d)^{-\frac{r}{r+1}}, \text{ if } N \geq \eta^{\frac{r}{r+1}} d^{\frac{2r+1}{r+1}}. \tag{3.7}$$

Notice that by the conditions in Proposition 3.1 and Theorem 3.2, all the above conclusions hold when $t = \widetilde{t} \cdot \sigma^{-2}$ is not exceeding the order of $\eta(r+1)^{(r+1)/2} n^{(r+1)/2}$, which, under the parameter regime (3.5), translates to $\widetilde{t} \lesssim d^{(r-1)/2}$. Thus we are able to observe that in the high-dimensional regime, vanilla GD method has the disadvantage of harmful overfitting to the label noise when the effective reasoning path length $\widetilde{t}$ is increasing, while the sampling-based test-time computing does not (see details in Remark D.4).

4 Analysis of In-context Sparse Linear Regression in Discrete Space

In this section, we conduct a theoretical analysis for binary sparse in-context linear regression (Section 2.3). Our strategy of studying and comparing the test-time computing methods is to analyze the probability of perfectly recovering the true coefficient, i.e., $\mathbb{P}(\mathbf{w}_t^{\mathtt{greedy}} = \mathbf{w}^*)$ and $\mathbb{P}(\mathbf{w}_{t,N}^{\mathtt{mv}} = \mathbf{w}^*)$. We use the notation $p(\mathbf{w}_t = \mathbf{w}) := \mathbb{P}(\mathbf{w}_t = \mathbf{w} \mid \mathbf{w}_0, \mathcal{D})$ to indicate the probability of weight $\mathbf{w}$ after t reasoning steps, conditioning on the initial state $\mathbf{w}_0$ and the in-context dataset $\mathcal{D}$ in a single reasoning path. We define $\mathcal{W} = \{\mathbf{w} \mid \mathbf{w} \in \{0,1\}^d, \|\mathbf{w}\|_0 = k\}$ and assume $\boldsymbol{x} \sim \mathcal{N}(0, \mathbf{I}_d)$ and label noise $\epsilon_i \sim \mathcal{N}(0, \sigma\epsilon^2)$ with $\sigma_\epsilon > 0$.

Our first result shows that if in a single reasoning path the prediction $\mathbf{w}_t$ has a probability of recovering the truth higher than that of recovering any other coefficient, then majority vote recovers the truth with a probability converging to 1 exponentially fast. The proof is in Appendix F.1.

Proposition 4.1 (Sample complexity for majority vote)**.** *Consider the binary sparse in-context linear regression task (Section 2.3) and using majority vote with reasoning length* T *and sampling number* N*. The final prediction* $\mathbf{w}_{t,N}^{\mathtt{mv}}$ *can asymptotically recover the truth* $\mathbf{w}^*$ *with probability* 1 *given sufficient sample size* N *if for a single reasoning path*

$$\Delta_t := p(\mathbf{w}_t = \mathbf{w}^*) - \max_{\mathbf{w}' \in \mathcal{W} \setminus \{\mathbf{w}^*\}} p(\mathbf{w}_t = \mathbf{w}') > 0. \tag{4.1}$$

Under condition (4.1)*, it holds that*

$$\mathbb{P}\left(\mathbf{w}_{t,N}^{\mathtt{mv}} = \mathbf{w}^* \mid \mathbf{w}_0, \mathcal{D}\right) \geq 1 - |\mathcal{W}| \cdot \exp\left(-N\Delta_t^2/2\right). \tag{4.2}$$

We remark that similar results of Proposition 4.1 have also been proposed in [52]. Here, we further provide more detailed analysis for the majority vote in our binary sparse linear regression task, show its dependence on the in-context example number n, reasoning length t, and compare it with the greedy decoding algorithm to emphasize when it is important to use the sample-then-select method.

Our main result to this end is the following two theorems. The first result is regarding the regime where we have sufficiently many in-context data n, with proof in Appendix F.2.

Theorem 4.2 (Perfect recovery probability with sufficient in-context examples)**.** *Suppose that* $n \geq (6k + 3\sigma_\epsilon)^4$*, then the overall recovery probability of greedy decoding and majority vote are lower bounded as following:*

- ***Greedy decoding:*** *for any reasoning length* $t \geq 1$, $\mathbb{P}\big(\mathbf{w}_t^{\mathtt{greedy}} = \mathbf{w}^*\big) \geq 1 - \delta(n)$
- ***Majority vote:*** *for any reasoning length* $t \geq 1$ *and sampling number* $N \geq 1$*, it holds that*

$$\mathbb{P}\left(\mathbf{w}_{t,N}^{\mathtt{mv}} = \mathbf{w}^*\right) \geq \big(1 - \delta(n)\big) \cdot \big(1 - |\mathcal{W}| \cdot e^{-N\Delta_t^2/2}\big). \tag{4.3}$$

Here $\delta(n) = 2d(d+2) \cdot \exp(-c \cdot n^{1/2})$ *for some absolute constant* $c > 0$*, and for any* $t \geq 1$*,* Δ_t *satisfies that*

$$\Delta_t \geq \frac{p_{\mathtt{trans}}}{p_{\mathtt{trans}} + 1 - p_{\mathtt{recurr}}} \left(1 - (p_{\mathtt{recurr}} - p_{\mathtt{trans}})^{t-1}\right), \tag{4.4}$$

where the quantities $p_{\mathtt{trans}}, p_{\mathtt{recurr}} \in (0,1)$ *are defined as*

$$p_{\mathtt{trans}} := \left(1 - \frac{2k + \sigma_\epsilon}{n^{1/4} - (2k + \sigma_\epsilon)}\right) \cdot \frac{1}{d^k}, \quad p_{\mathtt{recurr}} := \left(1 - \frac{\sigma_\epsilon}{n^{1/4} - \sigma_\epsilon}\right) \cdot \left(\frac{n^{1/4} - \sigma_\epsilon}{n^{1/4} - \sigma_\epsilon + d\sigma_\epsilon}\right)^k. \tag{4.5}$$

Theorem 4.2 establishes lower bounds on the recovery probability for both greedy decoding and majority vote. The recovery probability improves exponentially with the number of in-context examples. For majority vote, since $0 < \Delta_t < 1$ for all $t \geq 1$, as with sufficiently many number of sampling paths ($N \to \infty$), we have $\mathbb{P}\left(\mathbf{w}_{t,\infty}^{\mathtt{mv}} = \mathbf{w}^*\right) \geq 1 - \delta$, which matches that of greedy decoding $\mathbb{P}(\mathbf{w}_t^{\mathtt{greedy}} = \mathbf{w}^*)$, and both algorithms can achieve perfect accuracy given sufficient in-context examples n. Moreover, we remark that $p_{\mathtt{recurr}} > p_{\mathtt{trans}}$ since it holds that $(n^{1/4} - \sigma_\epsilon)(n^{1/4} - \sigma_\epsilon + d\sigma_\epsilon)^{-1} > d^{-1}$ for sufficiently many in-context examples $n > (3\sigma_\epsilon)^4$. When $\sigma_\epsilon = 0$, we have $p_{\mathtt{recurr}} = 1$ and $p_{\mathtt{trans}} > 1/2d^k$, ensuring that Δ_t converges to 1 as $t \to \infty$.

The theorem for sufficient in-context data does not highlight the advantage of majority vote in terms of recovery probability. However, real-world applications and our experiments show that majority vote is more accurate and robust with limited in-context data. We present our second main theorem to analyze this scenario, considering the case with only one in-context example ($n = 1$ and $k = 1$). Although simplified, this case offers valuable insights into the robustness of majority vote.

Theorem 4.3 (Majority vote outperforms greedy decoding in the case of limited in-context examples)**.** *Consider the case where* $n = k = 1, \sigma_\epsilon = 0$*, and denote the in-context example as* $(\mathbf{x}, \mathbf{x}^\top \mathbf{w}^*)$*. We have the following results.*

- *Greedy decoding: for any reasoning length* $t \geq 1$*,*

$$\mathbb{P}\big(\mathbf{w}_t^{\mathtt{greedy}} = \mathbf{w}^*\big) \leq \frac{1}{2^{d-1}} + \frac{2}{d}. \tag{4.6}$$

- *Majority vote: there exists a* $\zeta > 0$ *such that for reasoning steps* $t \geq 2\log 2/\log(1-\zeta)$*, sampling number* $N \geq 1$*,*

$$\mathbb{P}\left(\mathbf{w}_{t,N}^{\mathtt{mv}} = \mathbf{w}^*\right) \geq 1 - \frac{1}{2^{d-1}}. \tag{4.7}$$

Theorem 4.3, detailed with proof in Appendix F.3, highlights a key difference between majority vote and greedy decoding with limited in-context examples. As shown in numerical experiments, greedy decoding frequently gets stuck in cyclic state transitions, failing to reach the optimal state $\mathbf{w}^*$. In contrast, majority vote explores the state space more effectively, enabling a high probability of converging to $\mathbf{w}^*$ even in constrained scenarios, as shown in numerical experiments in Section 5.1.

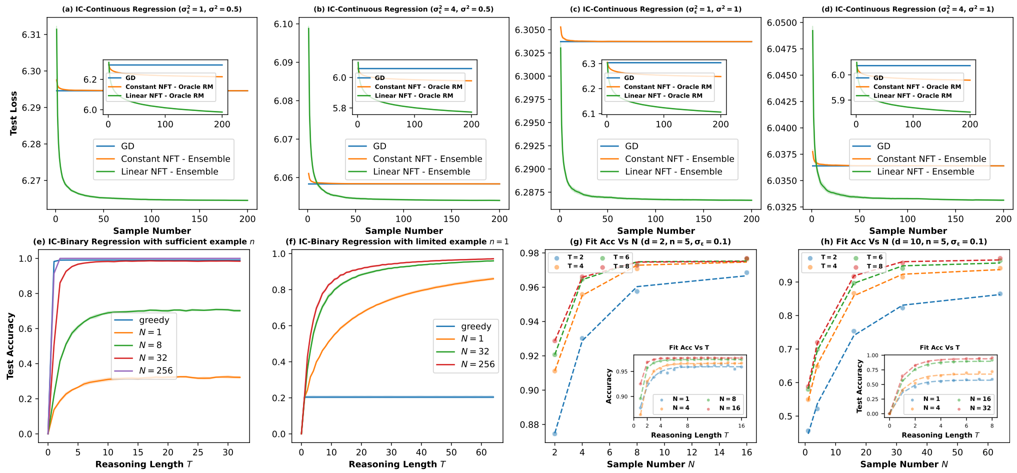

Figure 2: Numerical experiments on in-context linear regression with continuous coefficients (*a*-*d*) and binary coefficients (*e*-*h*).

5 Experiments

5.1 Numerical Results for In-Context Linear Regression

Here, we validate our theoretical findings through numerical experiments. For the continuous case, we examine the effects of varying σ_ϵ and σ. Our results demonstrate that with ensemble aggregation, constant NFT provides no performance improvement, while linear NFT reduces test loss given sufficient sample size, confirming Corollary 3.2. Furthermore, when decoding with a reward model, even constant NFT yields consistent performance improvements as sample numbers increase.

For the binary sparse coefficient case, we observe from Fig 2 (e) that with sufficient examples, both greedy decoding and majority voting achieve perfect accuracy, supporting Theorem 4.2. From Fig 2 (f) we find that when setting $n = 1$ and $d = 10, \sigma_\epsilon = 0$, with sufficiently large reasoning length T, majority voting achieves high accuracy, while greedy search maintains approximately $2/d = 0.2$ accuracy, consistent with Theorem 4.3. We fit the relationship between accuracy $\texttt{Acc}$ and sample number N using $\texttt{Acc} = \alpha_T - \beta_T e^{-\nu_T N}$ for given T. The results, shown in Fig 2 (g) and (h), not only validate Theorem 4.1 but also suggest practical applications for real-world LLM inference.

5.2 Insights for LLM Inference

Our theoretical analysis identifies two critical terms governing the model's behavior: $\mathcal{O}(e^{-\Delta_T^2 N/2})$ and $\mathcal{O}(e^{-\mu T})$, which determine the overall accuracy $\texttt{Acc}(T, N)$ and probability gap Δ_T. Leveraging these theoretical insights, we investigate real-world LLM inference behavior by developing a Low-Cost-to-High Prediction Algorithm (Algorithm 3; detailed in Appendix B.2). This algorithm successfully predicts model performance under computationally expensive settings using only data from configurations with relatively low reasoning tokens T or sampling numbers N, as illustrated in Fig. 5.2. The results demonstrate the potential of applying our theoretical framework for practical LLM behavior analysis.

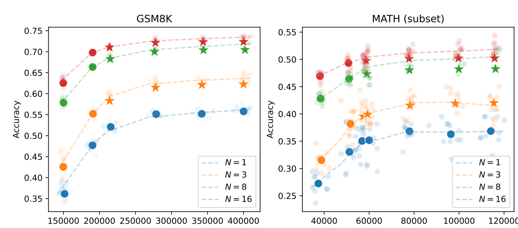

Figure 3: Utilizing data with low computational costs to forecast results for high computational costs, where ★ denotes predicted results and ● denotes the data utilized.

6 Conclusions and Limitations

This paper makes the initial step toward bridging the gap between practical language model test-time computing techniques with sampling and theoretical transformer analysis by incorporating randomness into the decoding process. We study the task of in-context linear regression with continuous/binary coefficients and provide a detailed analysis of widely adopted inference techniques, offering new insights into inference behaviors in real-world language models. Potential future works include analyzing other types of sampling algorithms and reasoning methods. Also it remains open to rigorously analyze the benefits of BoN method and its variants (with respect to different reward models) that we experimentally verified to be effective.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] Niket Agarwal, Arslan Ali, Maciej Bala, Yogesh Balaji, Erik Barker, Tiffany Cai, Prithvijit Chattopadhyay, Yongxin Chen, Yin Cui, Yifan Ding, et al. Cosmos world foundation model platform for physical ai. *arXiv preprint arXiv:2501.03575*, 2025.

[3] Kwangjun Ahn, Xiang Cheng, Hadi Daneshmand, and Suvrit Sra. Transformers learn to implement preconditioned gradient descent for in-context learning. *Advances in Neural Information Processing Systems*, 36:45614–45650, 2023.

[4] Kwangjun Ahn, Xiang Cheng, Minhak Song, Chulhee Yun, Ali Jadbabaie, and Suvrit Sra. Linear attention is (maybe) all you need (to understand transformer optimization). *arXiv preprint arXiv:2310.01082*, 2023.

[5] Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. What learning algorithm is in-context learning? investigations with linear models. In *The Eleventh International Conference on Learning Representations*, 2022.

[6] Yu Bai, Fan Chen, Huan Wang, Caiming Xiong, and Song Mei. Transformers as statisticians: Provable in-context learning with in-context algorithm selection. *Advances in neural information processing systems*, 36, 2024.

[7] Peter L Bartlett, Philip M Long, Gábor Lugosi, and Alexander Tsigler. Benign overfitting in linear regression. *Proceedings of the National Academy of Sciences*, 117(48):30063–30070, 2020.

[8] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[9] Siyu Chen, Heejune Sheen, Tianhao Wang, and Zhuoran Yang. Training dynamics of multi-head softmax attention for in-context learning: Emergence, convergence, and optimality. *arXiv preprint arXiv:2402.19442*, 2024.

[10] Xingwu Chen and Difan Zou. What can transformer learn with varying depth? case studies on sequence learning tasks. *arXiv preprint arXiv:2404.01601*, 2024.

[11] Xingwu Chen, Lei Zhao, and Difan Zou. How transformers utilize multi-head attention in in-context learning? a case study on sparse linear regression. *arXiv preprint arXiv:2408.04532*, 2024.

[12] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

[13] Hanze Dong, Wei Xiong, Deepanshu Goyal, Yihan Zhang, Winnie Chow, Rui Pan, Shizhe Diao, Jipeng Zhang, Kashun Shum, and Tong Zhang. Raft: Reward ranked finetuning for generative foundation model alignment. *arXiv preprint arXiv:2304.06767*, 2023.

[14] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[15] Deqing Fu, Tian-Qi Chen, Robin Jia, and Vatsal Sharan. Transformers learn higher-order optimization methods for in-context learning: A study with linear models. *arXiv preprint arXiv:2310.17086*, 2023.

[16] Shivam Garg, Dimitris Tsipras, Percy Liang, and Gregory Valiant. What Can Transformers Learn In-Context? A Case Study of Simple Function Classes, August 2023.

[17] Angeliki Giannou, Liu Yang, Tianhao Wang, Dimitris Papailiopoulos, and Jason D. Lee. How Well Can Transformers Emulate In-context Newton's Method?, March 2024.

[18] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[19] Tianyu Guo, Wei Hu, Song Mei, Huan Wang, Caiming Xiong, Silvio Savarese, and Yu Bai. How Do Transformers Learn In-Context Beyond Simple Functions? A Case Study on Learning with Representations, October 2023.

[20] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

[21] Xinyang Hu, Fengzhuo Zhang, Siyu Chen, and Zhuoran Yang. Unveiling the statistical foundations of chain-of-thought prompting methods. *arXiv preprint arXiv:2408.14511*, 2024.

[22] Jianhao Huang, Zixuan Wang, and Jason D. Lee. Transformers learn to implement multi-step gradient descent with chain of thought. In *The Thirteenth International Conference on Learning Representations*, 2025. URL `https://openreview.net/forum?id=r3DF5sOo5B`.

[23] Yu Huang, Yuan Cheng, and Yingbin Liang. In-Context Convergence of Transformers, October 2023.

[24] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[25] Junghwan Kim, Michelle Kim, and Barzan Mozafari. Provable memorization capacity of transformers. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=8JCg5xJCTPR`.

[26] Juno Kim and Taiji Suzuki. Transformers provably solve parity efficiently with chain of thought. *arXiv preprint arXiv:2410.08633*, 2024.

[27] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213, 2022.

[28] Vladimir Koltchinskii and Karim Lounici. Concentration inequalities and moment bounds for sample covariance operators. *Bernoulli*, pages 110–133, 2017.

[29] Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, John D Co-Reyes, Avi Singh, Kate Baumli, Shariq Iqbal, Colton Bishop, Rebecca Roelofs, et al. Training language models to self-correct via reinforcement learning. *arXiv preprint arXiv:2409.12917*, 2024.

[30] Fei Liu et al. Learning to summarize from human feedback. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020.

[31] Renpu Liu, Ruida Zhou, Cong Shen, and Jing Yang. On the learn-to-optimize capabilities of transformers in in-context sparse recovery. *arXiv preprint arXiv:2410.13981*, 2024.

[32] Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: iterative refinement with self-feedback (2023). *arXiv preprint arXiv:2303.17651*, 2023.

[33] Sadegh Mahdavi, Renjie Liao, and Christos Thrampoulidis. Memorization Capacity of Multi-Head Attention in Transformers, October 2023.

[34] Eran Malach. Auto-Regressive Next-Token Predictors are Universal Learners, September 2023.

[35] Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, et al. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*, 2021.

[36] OpenAI. Learning to reason with llms. *https://openai.com/index/learning-to-reason-with-llms/*, 2024.

[37] William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4195–4205, 2023.

[38] William Saunders, Catherine Yeh, Jeff Wu, Steven Bills, Long Ouyang, Jonathan Ward, and Jan Leike. Self-critiquing models for assisting human evaluators. *arXiv preprint arXiv:2206.05802*, 2022.

[39] Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024.

[40] Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM Test-Time Compute Optimally can be More Effective than Scaling Model Parameters, August 2024.

[41] Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021, 2020.

[42] Shokichi Takakura and Taiji Suzuki. Approximation and Estimation Ability of Transformers for Sequence-to-Sequence Functions with Infinite Dimensional Input, May 2023.

[43] Alexander Tsigler and Peter L Bartlett. Benign overfitting in ridge regression. *Journal of Machine Learning Research*, 24(123):1–76, 2023.

[44] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

[45] Roman Vershynin. *High-dimensional probability: An introduction with applications in data science*, volume 47. Cambridge university press, 2018.

[46] Johannes Von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent. In *International Conference on Machine Learning*, pages 35151–35174. PMLR, 2023.

[47] Johannes von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent, May 2023.

[48] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

[49] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-Consistency Improves Chain of Thought Reasoning in Language Models, March 2023.

[50] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[51] Kaiyue Wen, Huaqing Zhang, Hongzhou Lin, and Jingzhao Zhang. From sparse dependence to sparse attention: Unveiling how chain-of-thought enhances transformer sample efficiency. *arXiv preprint arXiv:2410.05459*, 2024.

[52] Yangzhen Wu, Zhiqing Sun, Shanda Li, Sean Welleck, and Yiming Yang. Inference Scaling Laws: An Empirical Analysis of Compute-Optimal Inference for Problem-Solving with Language Models, October 2024.

[53] Tong Yang, Yu Huang, Yingbin Liang, and Yuejie Chi. In-Context Learning with Representations: Contextual Generalization of Trained Transformers, August 2024.

[54] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36, 2024.

[55] Zhenrui Yue, Honglei Zhuang, Aijun Bai, Kai Hui, Rolf Jagerman, Hansi Zeng, Zhen Qin, Dong Wang, Xuanhui Wang, and Michael Bendersky. Inference scaling for long-context retrieval augmented generation. *arXiv preprint arXiv:2410.04343*, 2024.

[56] Hugh Zhang and Celia Chen. Test-time compute scaling laws. `https://github.com/hughbzhang/o1_inference_scaling_laws`, 2024.

[57] Ruiqi Zhang, Spencer Frei, and Peter L Bartlett. Trained transformers learn linear models in-context. *arXiv preprint arXiv:2306.09927*, 2023.

[58] Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Sun, Jeff Huang, Cody Hao Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E Gonzalez, et al. Sglang: Efficient execution of structured language model programs. *arXiv preprint arXiv:2312.07104*, 2024.

[59] Andy Zhou, Kai Yan, Michal Shlapentokh-Rothman, Haohan Wang, and Yu-Xiong Wang. Language agent tree search unifies reasoning acting and planning in language models. *arXiv preprint arXiv:2310.04406*, 2023.

[60] Difan Zou, Jingfeng Wu, Vladimir Braverman, Quanquan Gu, and Sham Kakade. Risk bounds of multi-pass sgd for least squares in the interpolation regime. *Advances in Neural Information Processing Systems*, 35:12909–12920, 2022.

A Related Works

Scaling test-time computing in LLMs. Scaling test-time computing has demonstrated tremendous empirical success in LLMs, especially for reasoning tasks [36]. Recent research on increasing test-time computing in LLMs primarily focuses on the following two aspects [39]: (i) generating longer reasoning paths, including chain-of-thought (CoT) prompting that elicits intermediate reasoning steps [50, 27] and self-refinement methods that iterate on previously generated content [32, 38, 29]; and (ii) generating multiple potential reasoning paths and selecting the optimal one through the methods such as consistency-based selection [49], reward-guided choosing [41, 30, 12, 13], reasoning tree search [54, 59], etc. Empirical studies demonstrate that increased test-time computation consistently improves model performance [40, 55, 36], suggesting the existence of inference scaling laws [52]. Nevertheless, the theoretical analysis of inference-time computing and its scaling law remains quite open.

Theory for transformer test-time computing. Inspired by the empirical success of the inference-time computing techniques of LLMs, recently there have been a few works trying to demystify the mechanism behind it through analysis on theoretical tasks and simple transformer models. Both [51, 26] consider how to train a one-layer transformer that utilizes CoT reasoning to efficiently solve the k-parity learning task, which provably improves over the same one without using CoT reasoning. [21] studies the statistical properties of CoT prompting and its variants including majority vote and tree-of-thought (ToT). However, their analysis is model agnostic and does not consider concrete transformer models compared with our work. The mostly related to our paper is the work of [**?**] who considers a one-layer transformer to solve in-context linear regression task with continuous coefficient. They show that the transformer can be well trained to perform vanilla multi-step GD with CoT. However, the fundamental difference between the study of [**?** 51, 26] and ours is that we propose to include randomness in the inference stage of the transformer models, which then allows us to go further and study more sophisticated test-time computing methods that involve randomly sampling multiple reasoning or CoT paths.

Theory for in-context learning by transformers. In-context learning (ICL) [8] is a key capability of LLMs which means that the model is able to answer a new query provided with a few query-answer demonstrations of the similar tasks without updating the model parameters. The empirical success of ICL methods has sparked a long line of theoretical research for the ICL ability of transformers. Most of these theoretical research builds on the in-context learning framework of [16], where input-output pairs are formalized as $\{(\mathbf{x}_i, f(\mathbf{x}_i))\}_{i=1}^{n}$, and the model (typically, transformers) is required to learn the unknown function $f(\cdot)$ from the context without updating the parameters. This framework enables theoretical analysis of transformers across multiple dimensions: expressive power [6, 19], mechanistic understanding [17, 47, 3], and training dynamics [57, 23, 9]. While most existing research treats transformer decoding as a deterministic process, theoretical understanding of test-time computation for transformer ICL remains in its infancy.

B Experiment Details

B.1 Experiment Settings

Details for Figure 1: We evaluate real-world LLM using the GSM8K dataset [12], employing SGLang [58] as our inference framework; and use synthetic data with our theoretical framework to simulate practical decoding procedures. The experimental configurations are as follows:

- **LLM Performance on GSM8K with Varying Sample Number**: We employ Llama3.1-8b[14] with an 8-shot chain-of-thought prompt following [50]. For each question, we generate 256 potential answers using decoding temperature of 1.0. We implement an oracle reward model that perfectly validates answer correctness, and set the temperature to 0.0 for greedy search.
- **LLM Performance on GSM8K with Varying Reasoning Lengths**: Using Llama3.1-8b-instruct, we analyze performance across different reasoning lengths, defined as the token consumption per inference call. Following [56], we incorporate token budgets into the prompts to constrain the model's responses. For each prompt, we generate 64 potential answers and create 10 random permutations of these answers. We define the reasoning length T as the sum of token consumption across all prompts, and for multiple samples ($N > 1$), we average the token counts over N. The accuracy-tokens curves are plotted using transparent scattered points for individual permutations and fitted with trend lines. The prompt templates are provided in G.
- **IC-Linear Regression with Continuous Coefficients**: We configure the parameters as $n = 36, d = 72, \eta = 1 \times 10^{-3}, \sigma_\epsilon^2 = 1, \sigma^2 = 4$, and present results at gradient descent iterations $t = 950$.
- **IC-Linear Regression with Binary Coefficients**: We set the parameters to $n = 4, k = 1, d = 48, \eta = \frac{1}{4}, \sigma_\epsilon^2 = 0.25$.

Details for Figure 2: we conduct numerical experiments on in-context linear regression with continuous coefficients (*above a-d*) and binary coefficients (*below e-h*), each setting we repeat 5 times, details are as follows:

- **Continuous case**: we set the parameters to $d = 72, n = 36, \eta = 10^{-3}$, and present results at gradient descent iterations $t = 950$.
- **Binary case**: In Figure 2 (e): we set $n = 40, k = 2, d = 30, \eta = \frac{1}{40}, \sigma_\epsilon = 0.1$; in (f): we set $n = 1, k = 1, d = 10, \eta = 1, \sigma_\epsilon = 0$; in (g): we set $n = 1, k = 1, d = 2, \eta = 1, \sigma_\epsilon = 0.1$; in (h): we set $n = 5, k = 1, d = 10, \eta = 1, \sigma_\epsilon = 0.1$.
- **Fitting accuracy with varying reasoning length** T: for $N = 1$, we fit the curve with

$$\texttt{Acc}(T, 1) \approx \alpha_1 - \beta_1 e^{-\mu_1 T},$$

  for $N > 1$, we first approximate $\Delta_T \approx \texttt{Acc}(T, 1) \approx \alpha_1 - \beta_1 e^{-\nu_1 T}$, where $(\alpha_1, \beta_1, \nu_1)$ are obtained in case $N = 1$, then fit curve with

$$\texttt{Acc}(T, N) \approx \alpha_N - \beta_N e^{-\mu_N \Delta_T^2}.$$

Details for Figure 5.2: We conduct experiments using GSM8K and a curated subset of the MATH dataset [20], details are as follows:

- **MATH Dataset Subset**: We filter the MATH to extract problems at level 1 with integer answers, yielding a subset of 309 problems.

- We maintain consistent experimental settings with the GSM8K reasoning length evaluation as in Figure 1, utilizing Llama3.1-8b-instruct with a decoding temperature of 1.0. To facilitate the fitting process in Algorithm 3, we apply a scaling factor of $\frac{1}{10^5}$ to the token count, $T' = \frac{T}{10^5}$.

B.2 Low-Cost-to-High Prediction algorithm

Our theoretical analysis reveals two critical terms $\mathcal{O}(e^{-\Delta_T^2 N/2})$ and $\mathcal{O}(e^{-\mu T})$ for the overall accuracy $\texttt{Acc}(T, N)$ and probability gap Δ_T. These findings can provide valuable insights into real-world LLM inference.

To begin, we can observe that Δ_T changes with the number of reasoning steps T in $\mathcal{O}(e^{-\mu T})$. This can be described as:

$$\Delta_T \approx \gamma - \kappa e^{-\mu T}. \tag{B.1}$$

Specifically, for sampling number of $N = 1$, here we *assume* we can directly express the overall accuracy as :

$$\texttt{Acc}(T, 1) \approx \gamma' - \kappa' e^{-\mu T}. \tag{B.2}$$

Note that Eq (B.2) and (B.1) shares the same μ. To predict the final accuracy for a given sampling number N, here we introduce two additional parameters $(\alpha_{(T,N)}, \beta_{(T,N)})$ and formulate $\texttt{Acc}(T, N)$ as:

$$\texttt{Acc}(T, N) \approx \alpha_{(T,N)} - \beta_{(T,N)} e^{-\Delta_T^2 N/2}. \tag{B.3}$$

To effectively fit Eq (B.1) - (B.3), based on the results on Fig 2 (g) and (h), we further claim two conjectures:

- When T is fixed, then Eq B.3 can be approximated by:

$$\texttt{Acc}(T, N) \approx \alpha_T - \beta_T e^{-\Delta_T^2 N/2}. \tag{B.4}$$

- When N is fixed, then Eq B.3 can be approximated by:

$$\texttt{Acc}(T, N) \approx \alpha_N - \beta_N e^{-\Delta_T^2 N/2}. \tag{B.5}$$

This analysis enables us to predict model's high test-time computation performance using data from low-computation, resulting our Low-Cost-to-High Prediction Algorithm 3:

Algorithm 3 Low-Cost-to-High Prediction algorithm

Part 1: Obtain (γ, κ, μ) in Eq B.1

1: **Input:** Data at varying cost $\{\texttt{Acc}^{(e)}(T_i, N_j)\}$,$T_i \in \mathcal{T}^{(e)}, N_j \in \mathcal{N}^{(e)}$;
2: $(\gamma', \kappa', \mu) \leftarrow$ Fit Eq B.2 with $\{\texttt{Acc}^{(e)}(T_i, 1)\}_{\mathcal{T}^{(e)}}$
3: $(\alpha_{T_1}, \beta_{T_1}, \Delta_{T_1}) \leftarrow$ Fit Eq B.4 with $\{\texttt{Acc}^{(e)}(T_1, N_j)\}$
4: $(\alpha_{T_2}, \beta_{T_2}, \Delta_{T_2}) \leftarrow$ Fit Eq B.4 with $\{\texttt{Acc}^{(e)}(T_2, N_j)\}$
5: $(\gamma, \kappa) \leftarrow$ Fit Eq B.1 with $\{(\Delta_{T_0}, \mu), (\Delta_{T_1}, \mu)\}$
6: **Return** (γ, κ, μ)

Part 2: Predict accuracy with (γ, κ, μ) and low cost data

1: **Input:** (γ, κ, μ) in Eq B.1,$\mathcal{D}_N = \{\texttt{Acc}^{(e)}(T_1, N), \texttt{Acc}^{(e)}(T_2, N)\}$;
2: $\Delta_{T_i} \leftarrow \gamma - \kappa e^{-\mu T_i}, i = 1, 2$ {//Eq B.1}
3: $(\alpha_N, \beta_N) \leftarrow$ Fit Eq B.5 with two data points: $\{(\texttt{Acc}^{(e)}(T_1, N), P_{T_1}), (\texttt{Acc}^{(e)}(T_2, N), P_{T_2})\}$
4: Use Eq B.1,Eq B.5 with obtained (γ, κ, μ) and (α_N, β_N) to predict data with varying T.

The core ideal of Algorithm 3 is to first determine (γ, κ, μ) in Equation B.1. Subsequently, we can compute Δ_T and Equation B.4 using two additional parameters α_N, β_N, obtainable from only two data points. Notably, since we use $\texttt{Acc}^{(e)}(T_0, N_j)$ and $\texttt{Acc}^{(e)}(T_1, N_j)$ during the initial parameter estimation (Algorithm 3 Part 1, lines 3-4), no additional data is required for subsequent predictions in part 2.

C Proofs for Section 2

C.1 Proof of Proposition 2.2

Proof of Proposition 2.2. The proof is based on the proof of Theorem 3.2 of [?]. We take the desired parameter $\theta_{\mathtt{GD}} = \{\mathbf{V}_{\mathtt{GD}}, \mathbf{W}_{\mathtt{GD}}\}$ as following,

$$\mathbf{V}_{\mathtt{GD}} := \begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 0 & \mathbf{0} & 0 \\ -\eta \cdot \mathbf{I}_d & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 0 & \mathbf{0} & 0 \end{pmatrix}, \quad \mathbf{W}_{\mathtt{GD}} := \begin{pmatrix} \mathbf{0} & \mathbf{0} & \mathbf{I}_d & \mathbf{0} \\ \mathbf{0} & 0 & \mathbf{0} & -1 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & 0 & \mathbf{0} & 0 \end{pmatrix}, \tag{C.1}$$

Then one can check that when inputting $\mathbf{H}_\ell$ in the form of

$$\mathbf{H}_\ell = \begin{pmatrix} \mathbf{x}_1 & \cdots & \mathbf{x}_n & \mathbf{0} & \cdots & \mathbf{0} \\ y_1 & \cdots & y_n & 0 & \cdots & 0 \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{w}_0 & \cdots & \mathbf{w}_\ell \\ 0 & \cdots & 0 & 1 & \cdots & 1 \end{pmatrix}, \tag{C.2}$$

the output embedding of the transformer at the last token is given by

$$(\widetilde{\mathbf{H}}_\ell)_{:,-1} = \begin{pmatrix} \mathbf{0} \\ 0 \\ \widetilde{\mathbf{w}}_\ell \\ 1 \end{pmatrix}, \quad \widetilde{\mathbf{w}}_\ell = \mathbf{w}_\ell - \frac{\eta}{n} \cdot \mathbf{X}^\top (\mathbf{X}\mathbf{w}_\ell - \mathbf{y}). \tag{C.3}$$

Thus if we take the sampling algorithm $\mathtt{Sampling\_Alg}(\cdot)$ satisfying the form of

$$\mathtt{Sampling\_Alg}(\mathbf{h}) = \delta_{\mathbf{0}}(\cdot) \otimes \delta_0(\cdot) \otimes p\left(\cdot | (\mathbf{h})_{d+2:2d+1}\right) \otimes \delta_1(\cdot), \tag{C.4}$$

for some conditional distribution $p : \mathbb{R}^d \mapsto \mathcal{P}(\mathbb{R}^d)$, then the embedding of the next token would be

$$\mathbf{h}_{\ell+1} = \begin{pmatrix} \mathbf{0} \\ 0 \\ \mathbf{w}_\ell \\ 1 \end{pmatrix}, \quad \mathbf{w}_{\ell+1} \sim p\left(\cdot \,\middle|\, \mathbf{w}_\ell - \frac{\eta}{n} \cdot \mathbf{X}^\top \big(\mathbf{X}\mathbf{w}_\ell - \mathbf{y}\big) \right), \tag{C.5}$$

by Definition 2.1. Iterating the above argument from $\ell = 0$ to $t-1$ completes the proof of Proposition 2.2. □

C.2 Special Case: Vanilla Multi-step Grandient Descent with CoT

One special case of Proposition 2.2 is a transformer that explicitly performs standard multi-step gradient descent (GD) [?], i.e., $p(\cdot|x) = \delta_x(\cdot)$, so that the final prediction of the regression coefficient after t reasoning steps is given by

$$\mathbf{w}_{\mathtt{GD}} := (\mathbf{H}_t)_{d+2:2d+1,n+t} = \left(\mathbf{I}_d - \left(\mathbf{I}_d - \frac{\eta}{n} \cdot \mathbf{X}^\top \mathbf{X} \right)^t \right) \mathbf{X}^\top (\mathbf{X}\mathbf{X}^\top)^{-1} \mathbf{y}. \tag{C.6}$$

We note that [?] considers transformer CoT reasoning for in-context-linear regression with *noiseless* labels, but here we allow the existence of label noise.

D Theoretical Analysis in Section 3 Continued

Theorem D.1 (Excess risk of vanilla multi-step GD with CoT: general covariance matrix)**.** *Under Assumption E.1, taking the step size $\eta \le \|\mathbf{H}\|_2^{-1}$ and CoT length t, with probability at least $1 - 1/\mathrm{poly}(n)$, it holds that*

$$\mathbb{E}_{\boldsymbol{\epsilon},\mathbf{w}^*}\left[\mathcal{E}(\mathbf{w}_{\mathtt{GD}})\right] \lesssim \omega^2 \cdot \left(\frac{\widetilde{\lambda}^2}{n^2} \cdot \sum_{1 \le i \le k^*} \frac{1}{\lambda_i} + \sum_{k^* < i \le d} \lambda_i \right) + \sigma_\epsilon^2 \cdot \left(\frac{k^*}{n} + \frac{n}{\widetilde{\lambda}^2} \cdot \sum_{k^* < i \le d} \lambda_i^2 \right), \tag{D.1}$$

where the quantities are as follows

$$k^* := \min\left\{ k : n\lambda_{k+1} \le \frac{n}{\eta t} + \sum_{k < i \le d} \lambda_i \right\}, \quad \widetilde{\lambda} := \frac{n}{\eta t} + \sum_{k^* < i \le d} \lambda_i. \tag{D.2}$$

Proof of Theorem D.1. Please refer to Appendix E.1 for a proof of Theorem D.1. □

Theorem D.2 (Excess risk of noisy multi-step noisy GD with CoT and ensembling)**.** *Under Assumption E.1, taking the step size $\eta \le \|\mathbf{H}\|_2^{-1}$ and CoT length t, we have the following risk bounds for* $\mathbf{w}_{\mathtt{avg}}$.

1. *Constant noise transformation function (Example 2.4): with probability at least $1 - 1/\mathrm{poly}(n)$,*

$$\mathbb{E}_{\boldsymbol{\epsilon},\mathbf{w}^*}\left[\mathcal{E}(\mathbf{w}_{\mathtt{GD}})\right] \lesssim \omega^2 \cdot \left(\frac{\widetilde{\lambda}^2}{n^2} \cdot \sum_{1\le i\le k^*} \frac{1}{\lambda_i} + \sum_{k^*<i\le d} \lambda_i\right) + \frac{\vartheta_{n,t}}{N}, \tag{D.3}$$

where the quantities k^* and $\widetilde{\lambda}$ are defined as the same as (D.2)*, and ϑ_t is defined as*

$$\vartheta_{n,t} := \sigma^2 d \cdot \left(t \cdot \sqrt{\frac{r(\mathbf{H}) \vee \log(\mathrm{poly(n)})}{n}} + \frac{1}{\eta}\right), \tag{D.4}$$

with $r(\mathbf{H}) = \mathrm{Tr}(\mathbf{H})/\|\mathbf{H}\|_2$ being the effective rank of $\mathbf{H}$.

2. *Linear noise transformation function (Example 2.5): taking the noise variance $\sigma^2 \asymp d^{-1}$ and the reasoning path length $t > \sigma^{-2} \cdot \log 2$, with probability at least $1 - 1/\mathrm{poly}(n)$,*

$$\mathbb{E}_{\boldsymbol{\epsilon},\mathbf{w}^*,\boldsymbol{\xi}}\left[\mathcal{E}(\mathbf{w}_{\mathtt{avg}})\right] \lesssim \omega^2 \cdot \left(\frac{(\widetilde{\lambda}^{\mathrm{Bias}})^2}{n^2} \cdot \sum_{1\le i\le k^*_{\mathrm{Bias}}} \frac{1}{\lambda_i} + \sum_{k^*_{\mathrm{Bias}}<i\le d} \lambda_i\right) + \sigma_\epsilon^2 \cdot \left(\frac{k^*_{\mathrm{Var}}}{n} + \frac{n}{(\widetilde{\lambda}^{\mathrm{Var}})^2} \cdot \sum_{k^*_{\mathrm{Var}}<i\le d} \lambda_i^2\right) + \frac{\varsigma_n}{N}, \tag{D.5}$$

where the quantities $\widetilde{\lambda}^{\mathrm{Bias}}$, $\widetilde{\lambda}^{\mathrm{Var}}$, k^*_{Bias}, and k^*_{Var} are defined as following respectively,

$$k^*_{(\diamondsuit)} := \min\left\{k \in [d] : n\lambda_{k+1} \le \widetilde{\lambda}^{(\diamondsuit)}_{\mathrm{effect}} + \sum_{k<i\le d} \lambda_i\right\}, \quad \widetilde{\lambda}^{(\diamondsuit)} := \widetilde{\lambda}^{(\diamondsuit)}_{\mathrm{effect}} + \sum_{k^*<i\le d} \lambda_i, \quad \text{for } (\diamondsuit) \in \{\mathrm{Bias}, \mathrm{Var}\}, \tag{D.6}$$

with $\widetilde{\lambda}^{\mathrm{Bias}}_{\mathrm{effect}}$ and $\widetilde{\lambda}^{\mathrm{Var}}_{\mathrm{effect}}$ defined as,

$$\widetilde{\lambda}^{\mathrm{Bias}}_{\mathrm{effect}} := \frac{n}{\eta} \cdot \left(\frac{2}{t} + \frac{\sigma^2}{1-\sigma^2}\left(1 + \frac{2}{t}\right)\right), \quad \widetilde{\lambda}^{\mathrm{Var}}_{\mathrm{effect}} := \frac{\sigma^2 n}{(1-\sigma^2)\eta}, \tag{D.7}$$

and the quantity ς_n, is given by

$$\varsigma_n := \left(\frac{\eta\sigma_\epsilon^2 d}{n\sigma^2} \cdot \mathrm{Tr}(\mathbf{H}) + \omega^2\right) \cdot \|\mathbf{H}\|_2. \tag{D.8}$$

Proof of Theorem D.2. Please refer to Appendix E.3 for a proof of Theorem D.2. □

Corollary D.3 (Theorem 3.2 restated)**.** *Under the same assumptions and setups as in Theorem D.2, additionally assuming that the spectrum of $\mathbf{H}$ satisfies polynomially decaying, i.e., $\lambda_i = i^{-(r+1)}$ for some constant $r \ge 0$, we have the following results.*

1. *Constant noise transformation function (Example 2.4): taking the reasoning path length $t \lesssim \eta(r+1)^{(r+2)/2} n^{(r+1)/2}$ and the sampling path number*

$$N \ge N_{\mathrm{c}} := \left(\sigma^2 d \cdot \left(t \cdot \sqrt{\frac{r(\mathbf{H}) \vee \log(\mathrm{poly(n)})}{n}} + \frac{1}{\eta}\right)\right) \cdot \left(\omega^2 \cdot \left(\frac{1}{t\eta}\right)^{\frac{r}{r+1}} + \frac{\sigma_\epsilon^2}{n} \cdot (t\eta)^{\frac{1}{r+1}}\right)^{-1}, \tag{D.9}$$

then with probability at least $1 - 1/\mathrm{poly}(n)$,

$$\mathbb{E}\left[\mathcal{E}(\mathbf{w}_{\mathtt{avg}})\right] \lesssim \omega^2 \cdot \left(\frac{1}{t\eta}\right)^{\frac{r}{r+1}} + \frac{\sigma_\epsilon^2}{n} \cdot (t\eta)^{\frac{1}{r+1}}. \tag{D.10}$$

2. *Linear noise transformation function (Example 2.5): taking the noise variance $\sigma^2 \asymp d^{-1}$, the reasoning path length $\sigma^{-2} \cdot \log 2 < t$, and the sampling path number*

$$N \geq N_l := \left(\omega^2 + \frac{\eta\sigma_\epsilon^2 d \cdot \mathrm{Tr}(\mathbf{H})}{n\sigma^2}\right) \cdot \|\mathbf{H}\|_2 \cdot \left(\omega^2 \cdot \left(\frac{\sigma^2}{\eta \cdot (1-\sigma^2)}\right)^{\frac{r}{r+1}} + \frac{\sigma_\epsilon^2}{n} \cdot \left(\frac{\eta \cdot (1-\sigma^2)}{\sigma^2}\right)^{\frac{1}{r+1}}\right)^{-1} \tag{D.11}$$

$$\asymp \left(\omega^2 + \frac{\sigma_\epsilon^2}{n} \cdot \eta d^2\right) \cdot \left(\omega^2 \cdot \left(\frac{1}{\eta d}\right)^{\frac{r}{r+1}} + \frac{\sigma_\epsilon^2}{n} \cdot (\eta d)^{\frac{1}{r+1}}\right)^{-1} \tag{D.12}$$

then with probability at least $1 - 1/\mathrm{poly}(n)$,

$$\mathbb{E}\left[\mathcal{E}(\mathbf{w}_{\mathtt{avg}})\right] \lesssim \omega^2 \cdot \widetilde{\lambda}^{\frac{r}{r+1}} + \frac{\sigma_\epsilon^2}{n} \cdot \left(\frac{\eta(1-\sigma^2)}{\sigma^2}\right)^{\frac{1}{r+1}}, \tag{D.13}$$

where $\widetilde{\lambda} := \eta^{-1}(2t^{-1} + \sigma^2(1+2t^{-1})/(1-\sigma^2))$.

Here the expectation is taken with respect to $\boldsymbol{\epsilon}$, $\mathbf{w}^*$, and the sampling noise $\boldsymbol{\xi}$ across different reasoning steps and paths.

Remark D.4. *Under the parameter regime of* (3.5)*, i.e.,*

$$\omega \asymp 1, \quad \sigma_\epsilon \asymp 1, \quad n \asymp \eta d, \quad \sigma^2 \asymp d^{-1}, \tag{D.14}$$

we can obtain further simplifications of the above result. Concretely, for the linear NFT setup, the number of sample paths needed is given by

$$N \geq N_l \asymp \left(\omega^2 + \sigma_\epsilon^2 d\right) \cdot \left(\left(\omega^2 + \sigma_\epsilon^2\right) \cdot \left(\frac{1}{\eta d}\right)^{\frac{r}{r+1}}\right)^{-1} \asymp d^{\frac{2r+1}{r+1}}, \tag{D.15}$$

and the excess risk bound is explicitly calculated by

$$\mathbb{E}_{\boldsymbol{\epsilon},\mathbf{w}^*,\boldsymbol{\xi}}\left[\mathcal{E}(\mathbf{w}_{\mathtt{avg,linear}})\right] \lesssim \left(\omega^2 + \sigma_\epsilon^2\right) \cdot (\eta d)^{-\frac{r}{r+1}} \asymp d^{-\frac{r}{r+1}}. \tag{D.16}$$

In contrast, we can also calculate that the risk bounds for either GD or ensemble with constant NFT is then given by

$$\mathbb{E}_{\boldsymbol{\epsilon},\mathbf{w}^*}\left[\mathcal{E}(\mathbf{w}_{\mathtt{GD}})\right], \mathbb{E}_{\boldsymbol{\epsilon},\mathbf{w}^*,\boldsymbol{\xi}}\left[\mathcal{E}(\mathbf{w}_{\mathtt{avg,const}})\right] \lesssim \widetilde{t}^{\frac{1}{r+1}} \cdot \left(\omega^2 + \sigma_\epsilon^2\right) \cdot (\eta d)^{-\frac{r}{r+1}} \asymp \widetilde{t}^{\frac{1}{r+1}} \cdot d^{-\frac{r}{r+1}}. \tag{D.17}$$

where $\widetilde{t} = \sigma^2 \cdot t$ is the scaled reasoning length, satisfying $\widetilde{t} \lesssim d^{(r-1)/2}$.

E Proofs for In-context Linear Regression with Continuous Coefficient (Section 3)

We denote the sample covariance matrix of the in-context data as $\boldsymbol{\Sigma} := n^{-1}\mathbf{X}^\top\mathbf{X} \in \mathbb{R}^{d\times d}$, and we define the gram matrix of the in-context data as $\mathbf{A} := \mathbf{X}\mathbf{X}^\top \in \mathbb{R}^{n\times n}$. Our results in this section depend on the following standard technical assumptions on the in-context data and task distributions.

Assumption E.1 (Data distribution)**.** *We assume the following on the in-context data distribution $\mathcal{D}_{\mathbf{w}^*}$:*

1. *The columns of $\mathbf{H}^{-1/2}\mathbf{x}$ are independent and* 1*-subGaussian;*

2. *The labels are generated according to $y = \mathbf{x}^\top\mathbf{w}^* + \epsilon$, where the label noise ϵ is independent of $\mathbf{x}$ and satisfies $\mathbb{E}[\epsilon] = 0$ and $\mathbb{E}[\epsilon^2] = \sigma_\epsilon^2$ for some constant $\sigma_\epsilon > 0$;*

3. *The true coefficient $\mathbf{w}^*$ follows the Gaussian prior, i.e., $\mathbf{w}^* \sim \mathcal{N}(\mathbf{0}, \omega^2 \cdot \mathbf{I}_d)$ for some constant $\omega > 0$.*

E.1 Proof of Theorem D.1

Proof of Theorem D.1. This follows from the same arguments as in the proof of Theorem 4.3 in [60]. We refer the readers to their proofs for seek of simplicity. ☐

E.2 Proof of Proposition 3.1

Proof of Proposition 3.1. As a special case of Theorem D.1, we begin by figuring out the optimal index k^*. We are going to prove that under the conditions in Proposition 3.1, the optimal index is given by

$$k^* = (\eta t)^{\frac{1}{r+1}} - 1. \tag{E.1}$$

Notice that here without loss of generality we assume that the above quantity is an integer since otherwise we can twist η (which is continuous) a little bit to make it an integer. And also we notice that the above $k^* \leq d$ due to our condition on t in Proposition 3.1. To prove this, it suffices to check that the above k^* is the smallest one satisfying the constraint in (D.2). To show it satisfies the constraint, consider

$$n\lambda_{k^*+1} = \frac{n}{(k^*+1)^{r+1}} = \frac{n}{\eta t} \leq \frac{n}{\eta t} + \sum_{k<i\leq d} \lambda_i. \tag{E.2}$$

To show that it is the smallest one satisfying the constraint, let's consider the other side of the inequality for $k^* - 1$. We have the following calculations. On the one hand, we have

$$n\lambda_{k^*} = \frac{n}{\left((\eta t)^{\frac{1}{r+1}} - 1\right)^{r+1}} = \frac{n}{\eta t} \cdot \frac{1}{\left(1 - (\eta t)^{-\frac{1}{r+1}}\right)^{r+1}} \geq \frac{n}{\eta t} \cdot \left(1 + (r+1) \cdot \left(\frac{1}{\eta t}\right)^{\frac{1}{r+1}}\right), \tag{E.3}$$

where the last inequality follows using $\log(1+x) \leq x$ and $\exp(x) \geq 1 + x$ to obtain the following argument

$$\frac{1}{\left(1 - (\eta t)^{-\frac{1}{r+1}}\right)^{r+1}} = \exp\left(-(r+1)\log\left(1 - (\eta t)^{-\frac{1}{r+1}}\right)\right) \geq \exp\left((r+1)(\eta t)^{-\frac{1}{r+1}}\right) \geq 1 + (r+1)(\eta t)^{-\frac{1}{r+1}}. \tag{E.4}$$

On the other hand, we have that

$$\frac{n}{\eta t} + \sum_{k^*-1<i\leq d} \lambda_i \leq \frac{n}{\eta t} + \sum_{i>k^*-1} \frac{1}{i^{r+1}} \leq \frac{n}{\eta t} + \frac{1}{\left((\eta t)^{\frac{1}{r+1}} - 1\right)^r} \lesssim \frac{n}{\eta t} + \left(\frac{1}{\eta t}\right)^{\frac{r}{r+1}}. \tag{E.5}$$

Now to see that $k^* - 1$ does not satisfies the constraint, in view of (E.3) and (E.5), it boils down to show that

$$\frac{n}{\eta t} \cdot \left(1 + (r+1) \cdot \left(\frac{1}{\eta t}\right)^{\frac{1}{r+1}}\right) \geq \frac{n}{\eta t} + \left(\frac{1}{\eta t}\right)^{\frac{r}{r+1}}, \tag{E.6}$$

which is equivalent to restricting the reasoning path length t satisfying $t \leq \eta \cdot (r+1)^{\frac{r+1}{2}} \cdot n^{\frac{r+1}{2}}$. According to our condition on the reasoning path length t in Proposition 3.1, this requirement does hold, and thus $k^* - 1$ does not satisfy the constraint. Therefore we have proved that $k^* = (\eta t)^{\frac{1}{r+1}} - 1$.

With the k^* in hand, we can then follow the same arguments as in the proof of Corollary 4.5 in [60] to obtain the final result. This completes the proof of Proposition 3.1. □

E.3 Proof of Theorem D.2

E.3.1 Proof for Example 2.4

Proof of Theorem D.2 for Example 2.4. Under this setting, each reasoning path is generated though the following iteration:

$$\mathbf{w}_{t+1}^{(j)} = \mathbf{w}_t^{(j)} - \frac{\eta}{n}\mathbf{X}^\top(\mathbf{X}\mathbf{w}_t^{(j)} - \mathbf{y}) + \boldsymbol{\xi}_t^{(j)}. \tag{E.7}$$

Based on this, we define the expected path $\mathbf{w}_t^{\mathtt{GD}(\eta;\mathbf{X},\mathbf{y})}$ and the fluctuation $\Delta_t^{(j)}$ iteratively as

$$\mathbf{w}_{t+1}^{\mathtt{GD}(\eta;\mathbf{X},\mathbf{y})} = \mathbf{w}_t^{\mathtt{GD}(\eta;\mathbf{X},\mathbf{y})} - \frac{\eta}{n}\mathbf{X}^\top(\mathbf{X}\mathbf{w}_t^{\mathtt{GD}(\eta;\mathbf{X},\mathbf{y})} - \mathbf{y}), \tag{E.8}$$

$$\Delta_{t+1}^{(j)} = \mathbf{w}_t^{(j)} - \mathbf{w}_t^{\mathtt{GD}(\eta;\mathbf{X},\mathbf{y})} \tag{E.9}$$

$$= (\mathbf{I} - \eta\mathbf{\Sigma})\Delta_t^{(j)} + \boldsymbol{\xi}_t^{(j)}. \tag{E.10}$$

By this characterization, we see that $\{\Delta_t^{(j)}\}_{j\le N}$ is a sequence of iid zero-mean random variable for fixed t. This expectation-fluctuation decomposition allows us to recast the risk of the sample averaged output as

$$\mathcal{E}(\mathbf{w}_t^{\mathtt{avg}}) = \mathcal{E}(\mathbf{w}_t^{\mathtt{GD}(\eta;\mathbf{X},\mathbf{y})}) + N^{-1}\mathbb{E}\big[\|\Delta_t^{(1)}\|_{\mathbf{H}}^2\big]. \tag{E.11}$$

In Theorem D.1, we have characterized the average-case risk of the gradient descent, therefore it suffices to study the fluctuation of a single reasoning path. In the sequel, we drop the superscript j for simplicity. Define $\mathbf{S}_t = \mathbb{E}[\Delta_t\Delta_t^\top]$, then we have that

$$\mathbf{S}_{t+1} = (\mathbf{I} - \eta\mathbf{\Sigma})\mathbf{S}_t(\mathbf{I} - \eta\mathbf{\Sigma})^\top + \sigma^2\mathbf{I} \tag{E.12}$$

$$= \sum_{j=0}^{t}\sigma^2(\mathbf{I} - \eta\mathbf{\Sigma})^{2j}, \tag{E.13}$$

where the last identity holds because of the deterministic initialization $\mathbf{S}_0 = 0$. Now we have that

$$\mathbb{E}[\|\Delta_t^{(j)}\|_{\mathbf{H}}^2] = \langle \mathbf{S}_t, \mathbf{\Sigma}\rangle + |\langle \mathbf{S}_t, \mathbf{H} - \mathbf{\Sigma}\rangle| \tag{E.14}$$

$$\le \mathrm{Tr}\Big(\sum_{j=0}^{t-1}\sigma^2(\mathbf{I} - \eta\mathbf{\Sigma})^{2j}\mathbf{\Sigma}\Big) + \mathrm{Tr}(\mathbf{S}_t)\cdot\|\mathbf{H} - \mathbf{\Sigma}\|_2. \tag{E.15}$$

For the first term above, we have that $\sum_{j=0}^{t}(1-\eta\lambda)^{2j}\lambda \le 1/\eta$ for $\lambda \in [0, 1/\eta]$. For the second term , we have by Koltchinskii and Lounici [28, Theorem 9] that there exists an event with probability $1-\delta$ over the randomness of $\mathbf{X}$, on which it holds that

$$\|\mathbf{H} - \mathbf{\Sigma}\|_2 \lesssim \sqrt{\frac{r(\mathbf{H}) \vee \log(1/\delta)}{n}}, \tag{E.16}$$

where $r(\mathbf{H}) = \mathrm{Tr}(\mathbf{H})/\|\mathbf{H}\|_2$ is the effective rank of $\mathbf{H}$. And we have the trivial upper bound that $\mathrm{Tr}(\mathbf{S}_t) \le \sigma^2 d\cdot t$. Plugging them into (E.11) and (E.15), we get that

$$\mathcal{E}(\mathbf{w}_t^{\mathtt{avg}}) \le \mathcal{E}(\mathbf{w}_t^{\mathtt{GD}(\eta;\mathbf{X},\mathbf{y})}) + N^{-1}\langle \mathbf{S}_t, \mathbf{H}\rangle \tag{E.17}$$

$$\le \mathcal{E}(\mathbf{w}_t^{\mathtt{GD}(\eta;\mathbf{X},\mathbf{y})}) + \frac{\sigma^2 d}{N}\Big(t\cdot\sqrt{\frac{r(\mathbf{H}) \vee \log(1/\delta)}{n}} + \frac{1}{\eta}\Big). \tag{E.18}$$

This concludes the proof of the theorem. □

E.3.2 Proof for Example 2.5

Now we give the proof of Theorem D.2 for Example 2.5. The proof relies on the following key lemmas.

Lemma E.2 (Error decomposition)**.** *The difference between $\mathbf{w}_{\mathtt{avg}}$ and the true coefficient $\mathbf{w}^*$ can be decomposed as following,*

$$\big\|\mathbf{w}_{\mathtt{avg}} - \mathbf{w}^*\big\|_{\mathbf{H}}^2 \le \mathrm{Bias} + \mathrm{Variance} + \mathrm{Fluctuation}, \tag{E.19}$$

where each of the three terms are defined as following,

$$\mathrm{Bias} := \big\|\big(\mathbf{X}^\top\mathbf{G}^{-1}\mathbf{X} - \mathbf{I}_d\big)\mathbf{w}^*\big\|_{\mathbf{H}}^2, \quad \mathrm{Variance} = \big\|\mathbf{X}^\top\mathbf{G}^{-1}\boldsymbol{\epsilon}\big\|_{\mathbf{H}}^2, \quad \mathrm{Fluctuation} = \Bigg\|\frac{1}{N}\sum_{j=1}^{N}\Delta^{(j)}\Bigg\|_{\mathbf{H}}^2, \tag{E.20}$$

with the matrix $\mathbf{G} \in \mathbb{R}^{n\times n}$ *and the vectors* $\{\Delta^{(j)}\}_{j=1}^N$ *defined as following,*

$$\mathbf{G} := \left(\frac{\sigma^2 n}{(1-\sigma^2)\eta}\cdot \mathbf{I}_n + \mathbf{A}\right)\left(\mathbf{I}_n - \big(1-\sigma^2\big)^t \cdot \left(\mathbf{I}_n - \frac{\eta}{n}\cdot \mathbf{A}\right)^t\right)^{-1}, \tag{E.21}$$

$$\Delta^{(j)} := \sum_{k=0}^{t-1}\left(\prod_{\ell=0}^{k-1}\big(\mathbf{I}_d - \boldsymbol{\xi}_{t-\ell}^{(j)}(\boldsymbol{\xi}_{t-\ell}^{(j)})^\top\big)\big(\mathbf{I}_d - \eta\boldsymbol{\Sigma}\big)\right)\big(\mathbf{I}_d - \boldsymbol{\xi}_{t-k}^{(j)}(\boldsymbol{\xi}_{t-k}^{(j)})^\top\big)\cdot\frac{\eta}{n}\cdot\mathbf{X}^\top\mathbf{y} \tag{E.22}$$

$$- \sum_{k=0}^{t-1}\big(1-\sigma^2\big)^{k+1}\big(\mathbf{I}_d - \eta\boldsymbol{\Sigma}\big)^k\cdot\frac{\eta}{n}\cdot\mathbf{X}^\top\mathbf{y}. \tag{E.23}$$

Proof of Lemma E.2. By definition, the output $\mathbf{w}_{\mathtt{avg}}$ is defined as

$$\mathbf{w}_{\mathtt{avg}} := \frac{1}{N}\sum_{j=1}^N \mathbf{w}_t^{(j)}, \tag{E.24}$$

where for each $j \in [N]$, the coefficient $\mathbf{w}_t^{(j)}$ is given by

$$\mathbf{w}_t^{(j)} = \sum_{k=0}^{t-1}\left(\prod_{\ell=0}^{k-1}\big(\mathbf{I}_d - \boldsymbol{\xi}_{t-\ell}^{(j)}(\boldsymbol{\xi}_{t-\ell}^{(j)})^\top\big)\big(\mathbf{I}_d - \eta\boldsymbol{\Sigma}\big)\right)\big(\mathbf{I}_d - \boldsymbol{\xi}_{t-k}^{(j)}(\boldsymbol{\xi}_{t-k}^{(j)})^\top\big)\cdot\frac{\eta}{n}\cdot\mathbf{X}^\top\mathbf{y} \tag{E.25}$$

$$= \Delta^{(j)} + \underbrace{\sum_{k=0}^{t-1}\big(1-\sigma^2\big)^{k+1}\big(\mathbf{I}_d - \eta\boldsymbol{\Sigma}\big)^k\cdot\frac{\eta}{n}\cdot\mathbf{X}^\top\mathbf{y}}_{:=\mathbf{w}_t}. \tag{E.26}$$

Now we decompose the difference between $\mathbf{w}_{\mathtt{avg}}$ in (E.24) and the truth $\mathbf{w}^*$ as following, considering

$$\mathbf{w}_{\mathtt{avg}} - \mathbf{w}^* = \frac{1}{N}\sum_{j=1}^N \mathbf{w}_t^{(j)} - \mathbf{w}^* = \mathbf{w}_t - \mathbf{w}^* + \frac{1}{N}\sum_{j=1}^N \Delta^{(j)}, \tag{E.27}$$

where the difference $\mathbf{w}_t - \mathbf{w}^*$ can be further explicitly expanded as

$$\mathbf{w}_t - \mathbf{w}^* = \sum_{k=0}^{t-1}\big(1-\sigma^2\big)^{k+1}\big(\mathbf{I}_d - \eta\boldsymbol{\Sigma}\big)^k\cdot\frac{\eta}{n}\cdot\mathbf{X}^\top\mathbf{y} - \mathbf{w}^* \tag{E.28}$$

$$= \sum_{k=0}^{t-1}\big(1-\sigma^2\big)^{k+1}\big(\mathbf{I}_d - \eta\boldsymbol{\Sigma}\big)^k\cdot\frac{\eta}{n}\cdot\mathbf{X}^\top\big(\mathbf{W}\mathbf{w}^* + \boldsymbol{\epsilon}\big) - \mathbf{w}^* \tag{E.29}$$

$$= \big(1-\sigma^2\big)\cdot\Big(\mathbf{I}_d - \big(1-\sigma^2\big)^t\big(\mathbf{I}_d - \eta\boldsymbol{\Sigma}\big)^t\Big)\Big(\sigma^2\mathbf{I}_d + \big(1-\sigma^2\big)\eta\boldsymbol{\Sigma}\Big)^{-1}\cdot\frac{\eta}{n}\cdot\mathbf{X}^\top\mathbf{X}\mathbf{w}^* - \mathbf{w}^* \tag{E.30}$$

$$+ \big(1-\sigma^2\big)\cdot\Big(\mathbf{I}_d - \big(1-\sigma^2\big)^t\big(\mathbf{I}_d - \eta\boldsymbol{\Sigma}\big)^t\Big)\Big(\sigma^2\mathbf{I}_d + \big(1-\sigma^2\big)\eta\boldsymbol{\Sigma}\Big)^{-1}\cdot\frac{\eta}{n}\cdot\mathbf{X}^\top\mathbf{X}\boldsymbol{\epsilon} \tag{E.31}$$

$$= \Big(\mathbf{X}^\top\mathbf{G}^{-1}\mathbf{X} - \mathbf{I}_d\Big)\mathbf{w}^* + \mathbf{X}^\top\mathbf{G}^{-1}\boldsymbol{\epsilon}, \tag{E.32}$$

where the last equality uses the definition of the matrix $\mathbf{G}$ in (E.21) and the fact that

$$\Big(\mathbf{I}_d - \big(1-\sigma^2\big)^t\big(\mathbf{I}_d - \eta\boldsymbol{\Sigma}\big)^t\Big)\Big(\sigma^2\mathbf{I}_d + \big(1-\sigma^2\big)\eta\boldsymbol{\Sigma}\Big)^{-1}\mathbf{X}^\top \tag{E.33}$$

$$= \mathbf{X}^\top\left(\mathbf{I}_n - \big(1-\sigma^2\big)^t\left(\mathbf{I}_d - \frac{\eta}{n}\mathbf{A}\right)^t\right)\Big(\sigma^2\mathbf{I}_n + \big(1-\sigma^2\big)\eta\mathbf{A}\Big)^{-1}. \tag{E.34}$$

Finally, by combining (E.27) and (E.32), we can arrive at

$$\big\|\mathbf{w}_{\mathtt{avg}} - \mathbf{w}^*\big\|_{\mathbf{H}}^2 = \left\|\Big(\mathbf{X}^\top\mathbf{G}^{-1}\mathbf{X} - \mathbf{I}_d\Big)\mathbf{w}^* + \mathbf{X}^\top\mathbf{G}^{-1}\boldsymbol{\epsilon} + \frac{1}{N}\sum_{j=1}^N\Delta^{(j)}\right\|_{\mathbf{H}}^2 \le \mathrm{Bias} + \mathrm{Variance} + \mathrm{Fluctuation}. \tag{E.35}$$

This completes the proof of Lemma E.2. □

Lemma E.3. *The matrix* $\mathbf{G}$ *satisfies the that for any CoT length* $t \geq \sigma^{-2} \cdot \log 2$*, it holds that*

$$\frac{\sigma^2 n}{(1-\sigma^2)\eta} \cdot \mathbf{I}_n + \mathbf{A} \preceq \mathbf{G} \preceq \frac{n}{\eta} \cdot \left(\frac{2}{t} + \frac{\sigma^2}{1-\sigma^2}\left(1 + \frac{2}{t}\right)\right) \cdot \mathbf{I}_n + \mathbf{A}. \tag{E.36}$$

Proof of Lemma E.3. It is direct from the definition of $\mathbf{G}$ in (E.21) to see the left side of the inequality. To prove the right side of the inequality, consider that by (E.21), we have the following,

$$\mathbf{G} - \left(\frac{\sigma^2 n}{(1-\sigma^2)\eta} \cdot \mathbf{I}_n + \mathbf{A}\right) \tag{E.37}$$

$$= \left(1-\sigma^2\right)^t \cdot \left(\frac{\sigma^2 n}{(1-\sigma^2)\eta} \cdot \mathbf{I}_n + \mathbf{A}\right)\left(\mathbf{I}_n - \frac{\eta}{n} \cdot \mathbf{A}\right)^t \left(\mathbf{I}_n - \left(1-\sigma^2\right)^t \cdot \left(\mathbf{I}_n - \frac{\eta}{n} \cdot \mathbf{A}\right)^t\right)^{-1}. \tag{E.38}$$

To proceed, it suffices to consider the real-valued single-variable function f defined as

$$f(x) = \frac{\left(\eta^{-1}\left(1-\sigma^2\right)^{-1} n\sigma^2 + x\right) \cdot \left(1 - n^{-1}\eta x\right)^t}{1 - \left(1-\sigma^2\right)^t \cdot \left(1 - n^{-1}\eta x\right)^t}. \tag{E.39}$$

On the one hand, for $t \geq \sigma^{-2} \cdot \log 2$, we have $t > -\log 2/\log(1-\sigma^2)(1-n^{-1}\eta x)$, and thus

$$1 - \left(1-\sigma^2\right)^t \cdot \left(1 - n^{-1}\eta x\right)^t \geq \frac{1}{2}. \tag{E.40}$$

On the other hand, by direct calculations we can see that the numerator is upper bounded by

$$\left(\frac{\sigma^2 n}{(1-\sigma^2)\eta} + x\right) \cdot \left(1 - n^{-1}\eta x\right)^t \leq \frac{1}{t} \cdot \frac{n}{\eta} \cdot \left(\frac{\sigma^2}{1-\sigma^2} + 1\right). \tag{E.41}$$

Consequently, by combining (E.40) and (E.41), we can see that for $t \geq \sigma^{-2} \cdot \log 2$,

$$f(x) \leq \frac{2}{t} \cdot \frac{n}{\eta} \cdot \left(\frac{\sigma^2}{1-\sigma^2} + 1\right), \tag{E.42}$$

which, combined with (E.37), further indicates that

$$\mathbf{G} - \left(\frac{\sigma^2 n}{(1-\sigma^2)\eta} \cdot \mathbf{I}_n + \mathbf{A}\right) \preceq \frac{2}{t} \cdot \frac{n}{\eta} \cdot \left(\frac{\sigma^2}{1-\sigma^2} + 1\right) \cdot \mathbf{A}. \tag{E.43}$$

This completes the proof of the right side inequality of Lemma E.3 and finishes the proof. □

Lemma E.4 (Bias error)**.** *Under Assumption E.1, taking the step size* $\eta \lesssim \mathrm{Tr}(\mathbf{H})^{-1}$ *and for any* $k \in [d]$*, with probability at least* $1 - 1/\mathrm{poly}(n)$*, it holds that*

$$\mathbb{E}_{\mathbf{w}^*}[\mathrm{Bias}] \lesssim \omega^2 \cdot \left(\left(\frac{1}{n^2} \cdot \left(\frac{n}{\eta} \cdot \left(\frac{2}{t} + \frac{\sigma^2}{1-\sigma^2} \cdot \left(1 + \frac{2}{t}\right)\right) + \sum_{k<i\leq d} \lambda_i\right)^2 \cdot \sum_{1\leq i\leq k} \frac{1}{\lambda_i} + \sum_{k<i\leq d} \lambda_i\right). \tag{E.44}$$

Proof of Lemma E.4. According to the definition of Bias in (E.20), using that $\mathbf{w}^* \sim \mathcal{N}(\mathbf{0}, \omega^2 \cdot \mathbf{I}_d)$ we have

$$\mathbb{E}_{\mathbf{w}^*}[\mathrm{Bias}] = \mathbb{E}_{\mathbf{w}^* \sim \mathcal{N}(\mathbf{0}, \omega^2 \cdot \mathbf{I}_d)}\left[\left\|\mathbf{H}^{\frac{1}{2}}\left(\mathbf{I}_d - \mathbf{X}^\top \mathbf{G}^{-1}\mathbf{X}\right)\mathbf{w}^*\right\|_2^2\right] \tag{E.45}$$

$$= \omega^2 \cdot \mathrm{Tr}\left(\mathbf{H}\left(\mathbf{I}_d - \mathbf{X}^\top \mathbf{G}^{-1}\mathbf{X}\right)^2\right) \tag{E.46}$$

$$\leq \omega^2 \cdot \mathrm{Tr}\left(\mathbf{H}\left(\mathbf{I}_d - \mathbf{X}^\top \left(\frac{n}{\eta} \cdot \left(\frac{2}{t} + \frac{\sigma^2}{1-\sigma^2}\left(1 + \frac{2}{t}\right)\right) \cdot \mathbf{I}_n + \mathbf{A}\right)^{-1} \mathbf{X}\right)^2\right), \tag{E.47}$$

where the last inequality follows from Lemma E.3. Notice that the quantity of trace on the right hand side actually corresponds to the bias error of the standard ridge regression with regularization coefficient $\widetilde{\lambda}_{\text{effect}}$ of

$$\widetilde{\lambda}_{\text{effect}}^{\text{Bias}} := \frac{n}{\eta} \cdot \left(\frac{2}{t} + \frac{\sigma^2}{1-\sigma^2} \left(1 + \frac{2}{t} \right) \right). \tag{E.48}$$

Thus by invoking Theorem 1 of [43], we can then obtain the result in Lemma E.4. □

Lemma E.5 (Variance error)**.** *Under Assumption E.1, taking the step size* $\eta \lesssim \mathrm{Tr}(\mathbf{H})^{-1}$ *and for any* $k \in [d]$*, with probability at least* $1 - 1/\mathrm{poly}(n)$*, it holds that*

$$\mathbb{E}_{\boldsymbol{\epsilon}}\left[\mathrm{Variance}\right] \lesssim \sigma_\epsilon^2 \cdot \left(\frac{k}{n} + n \cdot \left(\frac{\sigma^2 n}{(1-\sigma^2)\eta} + \sum_{k<i\le d} \lambda_i \right)^{-2} \cdot \sum_{k<i\le d} \lambda_i^2 \right). \tag{E.49}$$

Proof of Lemma E.5. According to the definition of Bias in (E.20), using that $\epsilon_i \sim \mathcal{N}(0, \sigma_\epsilon^2)$ we have

$$\begin{aligned}
\mathbb{E}_{\boldsymbol{\epsilon}}\left[\mathrm{Variance}\right] &= \mathbb{E}_{\epsilon\sim\mathcal{N}(\mathbf{0},\sigma_\epsilon^2\cdot\mathbf{I}_d)}\left[\left\| \mathbf{H}^{\frac{1}{2}} \mathbf{X}^\top \mathbf{G}^{-1} \boldsymbol{\epsilon} \right\|_2^2 \right] && \text{(E.50)}\\
&= \sigma_\epsilon^2 \cdot \mathrm{Tr}\left(\mathbf{X}\mathbf{H}\mathbf{X}^\top \mathbf{G}^{-2} \right) && \text{(E.51)}\\
&\le \sigma_\epsilon^2 \cdot \mathrm{Tr}\left(\mathbf{X}\mathbf{H}\mathbf{X}^\top \left(\frac{\sigma^2 n}{(1-\sigma^2)\eta} \cdot \mathbf{I}_n + \mathbf{A} \right)^{-2} \right) && \text{(E.52)}
\end{aligned}$$

Similar to the proof of Lemma E.4, the above quantity on the right hand side actually corresponds to the variance error of standard ridge regression with regularization coefficient $\widetilde{\lambda}_{\text{effect}}$ of

$$\widetilde{\lambda}_{\text{effect}}^{\text{Var}} := \frac{\sigma^2 n}{(1-\sigma^2)\eta}. \tag{E.53}$$

Consequently, by Theorem 1 of [43], we can obtain the result in Lemma E.5. □

Lemma E.6 (Fluctuation error)**.** *Suppose that we choose* $\sigma^2 < 1/(d+1)$ *and the step size* $\eta \lesssim \mathrm{Tr}(\mathbf{H})^{-1}$*. Then there exists an event with probability* $1 - 1/\mathrm{poly}(n)$ *over the randomness of* $\boldsymbol{X}$ *on which it holds that*

$$\mathbb{E}_{\mathbf{w}^*,\boldsymbol{\xi},\boldsymbol{\epsilon}}[\mathrm{Fluctuation}] \lesssim \frac{(\eta\sigma^{-2}\sigma_\epsilon^2 d \cdot \mathrm{Tr}(\mathbf{H})/n + \omega^2)\cdot \|\mathbf{H}\|_2}{N} \tag{E.54}$$

Proof of Lemma E.6. In the proof, we replace the notation $\Delta^{(j)}$ with Δ_t^j to emphasize the dependence on the reasoning step. From the characterization in Lemma E.2, we have for each path and its expectation over $\boldsymbol{\xi}$, it holds that

$$\begin{aligned}
\mathbf{w}_{t+1}^{(j)} &= (\mathbf{I} - \boldsymbol{\xi}_{t+1}^{(j)} \boldsymbol{\xi}_{t+1}^{(j)\top})(\mathbf{I} - \eta\boldsymbol{\Sigma})\left(\mathbf{w}_t^{(j)} + \eta\mathbf{X}^\top\mathbf{y}/n\right) && \text{(E.55)}\\
&= (1-\sigma^2)\cdot(\mathbf{I} - \eta\boldsymbol{\Sigma})(\mathbf{w}_t^{(j)} + \eta\mathbf{X}^\top\mathbf{y}/n) + \sigma^2 \cdot \left(\mathbf{I} - \sigma^{-2}\boldsymbol{\xi}_{t+1}^{(j)} \boldsymbol{\xi}_{t+1}^{(j)\top} \right)(\mathbf{w}_t^{(j)} + \eta\mathbf{X}^\top\mathbf{y}/n) && \text{(E.56)}\\
\mathbf{w}_{t+1} &= (1-\sigma^2)(\mathbf{I} - \eta\boldsymbol{\Sigma})(\mathbf{w}_t + \eta\mathbf{X}^\top\mathbf{y}/n). && \text{(E.57)}
\end{aligned}$$

Since there exists an event with probability $1 - 1/\mathrm{poly}(n)$ on which $\mathrm{Tr}(\boldsymbol{\Sigma}) \gtrsim \mathrm{Tr}(\mathbf{H})$, we have that $\eta < 1/\mathrm{Tr}(\boldsymbol{\Sigma})$ with high probability. In order to control the fluctuation error, we begin with deriving a deterministic upper bound on $\mathbf{w}_t$.

Bounding the expected path. By (E.57), the quantity $\mathbf{g}_t = \mathbf{w}_t + \eta \mathbf{X}^\top \mathbf{y}$ can be iteratively characterized as follows:

$$\mathbf{g}_{t+1} = (1-\sigma^2)(\mathbf{I} - \eta\boldsymbol{\Sigma})\mathbf{g}_t + \eta\mathbf{X}^\top\mathbf{y}/n \tag{E.58}$$
$$= \sum_{k=0}^{t}(1-\sigma^2)^k\big(\mathbf{I}-\eta\boldsymbol{\Sigma}\big)^k\eta\mathbf{X}^\top\mathbf{y}/n \tag{E.59}$$
$$= \sum_{k=0}^{t}\big(\mathbf{I}-\sigma^2\mathbf{I}-\eta\boldsymbol{\Sigma}+\eta\sigma^2\boldsymbol{\Sigma})^k\eta\boldsymbol{\Sigma}\mathbf{w}^* \tag{E.60}$$
$$+ \sum_{k=0}^{t}\big(\mathbf{I}-\sigma^2\mathbf{I}-\eta\boldsymbol{\Sigma}+\eta\sigma^2\boldsymbol{\Sigma})^k\eta\mathbf{X}^\top\boldsymbol{\epsilon}/n, \tag{E.61}$$

To this end, we define $p(z) = \sum_{k=0}^{t}(1-\sigma^2-z+\sigma^2 z)^k$. We can bound the scalar polynomials $p(z)$, $p(z)\cdot z$ and $p^2(z)\cdot z$ on $[0,1)$ as

$$p(z) \le \frac{1}{\sigma^2+(1-\sigma^2)z}; \tag{E.62}$$
$$p(z)\cdot z \le \frac{z}{\sigma^2+(1-\sigma^2)z} \lesssim (\sigma^{-2}z)\wedge 1; \tag{E.63}$$
$$p^2(z)\cdot z \le \frac{z}{\big(\sigma^2+(1-\sigma^2)z\big)^2} \lesssim (\sigma^{-4}z)\wedge z^{-1}. \tag{E.64}$$

We begin with the first term. It follows from (E.63) that

$$\|p(\eta\boldsymbol{\Sigma})\cdot\eta\boldsymbol{\Sigma}\|_2 \lesssim (\sigma^{-2}\cdot\eta\|\boldsymbol{\Sigma}\|_2)\wedge 1. \tag{E.65}$$

Therefore the first term can be upper bounded by $\big((\sigma^{-2}\eta\|\boldsymbol{\Sigma}\|_2)\wedge 1\big)\cdot\|\mathbf{w}^*\|_2$. For the second term, we have that

$$\mathbb{E}_{\boldsymbol{\epsilon}}\Big[\Big\|\sum_{k=0}^{t}\big(\mathbf{I}-\sigma^2\mathbf{I}-\eta\boldsymbol{\Sigma}+\eta\sigma^2\boldsymbol{\Sigma})^k\eta\mathbf{X}^\top\boldsymbol{\epsilon}/n\Big\|_2^2\Big] = \frac{\eta\sigma_\epsilon^2}{n}\cdot\mathrm{Tr}\Big(p(\eta\boldsymbol{\Sigma})\cdot\eta\boldsymbol{\Sigma}\cdot p(\eta\boldsymbol{\Sigma})\Big), \tag{E.66}$$

And therefore we have by (E.64) that

$$\mathbb{E}_{\boldsymbol{\epsilon},\mathbf{w}^*}[\sup_{t\ge 0}\|\mathbf{g}_t\|_2^2] \lesssim \frac{\eta\sigma_\epsilon^2}{n}\cdot\sigma^{-4}\mathrm{Tr}(\boldsymbol{\Sigma}) + \big(1\wedge\sigma^{-2}\eta\|\boldsymbol{\Sigma}\|_2\big)\|\mathbf{w}^*\|_2^2 \tag{E.67}$$
$$\lesssim \frac{\eta\sigma_\epsilon^2}{n}\sigma^{-4}\mathrm{Tr}(\boldsymbol{\Sigma}) + \|\mathbf{w}^\star\|_2^2. \tag{E.68}$$

Bounding the fluctuation. In the following, we use $\boldsymbol{\Lambda}_t^{(j)} = (\mathbf{I} - \sigma^{-2}\boldsymbol{\xi}_{t+1}^{(j)}\boldsymbol{\xi}_{t+1}^{(j)\top})$ for abbreviation. The fluctuation term $\Delta_t^{(j)}$ follows that

$$\Delta_{t+1}^{(j)} = \mathbf{w}_{t+1}^{(j)} - \mathbf{w}_{t+1} \tag{E.69}$$
$$= (1-\sigma^2)\cdot(\mathbf{I}-\eta\boldsymbol{\Sigma})\cdot\Delta_t^{(j)} + \sigma^2\cdot\boldsymbol{\Lambda}_t^{(j)}\cdot(\mathbf{w}_t^{(j)}+\eta\mathbf{X}^\top\mathbf{y}). \tag{E.70}$$

For each t, we have that $\boldsymbol{\Lambda}_t^{(j)}$ is independent with $\mathbf{w}_t^{(j)}$ and is of zero mean. Consequently we have that $\mathbb{E}[\Delta_t^{(j)}] = 0$ for any $t \ge 0$. Besides, it can be easily verified by induction that $\Delta_t^{(j)}, j \le N$ are independent and identically distributed. Thanks to this, we have that

$$\mathbb{E}\Big[\Big\|N^{-1}\sum_{j\le N}\Delta_t^{(j)}\Big\|_{\mathbf{H}}^2\Big] = \mathbb{E}\Big[N^{-2}\sum_{j\le N}\Delta_t^{(j)\top}\mathbf{H}\Delta_t^{(j)} + N^{-2}\sum_{j<k}\Delta_t^{(j)\top}\mathbf{H}\Delta_t^{(k)}\Big] \tag{E.71}$$
$$= N^{-1}\langle\mathbf{H}, \mathbb{E}[\Delta_t^{(j)\top}\Delta_t^{(j)}]\rangle. \tag{E.72}$$

Therefore, it suffices to upper bound the second moment of the fluctuation along a single reasoning path. For simplicity, let us drop the superscript (j) in the subsequent analysis. We study the iteration of the second moment $\mathbf{S}_t = \mathbb{E}[\Delta_t\Delta_t^\top]$. Rewriting (E.70), we get that

$$\Delta_{t+1} = (1-\sigma^2)\cdot(\mathbf{I}-\eta\boldsymbol{\Sigma})\Delta_t + \sigma^2\boldsymbol{\Lambda}_t\Delta_t \tag{E.73}$$
$$+ \sigma^2\boldsymbol{\Lambda}_t\cdot(\mathbf{w}_t+\eta\mathbf{X}^\top\mathbf{y}). \tag{E.74}$$

Note that $\mathbf{\Lambda}_t$ and Δ_t are zero mean and independent, we have that

$$\mathbf{S}_{t+1} = (1-\sigma^2)^2 \cdot (\mathbf{I} - \eta\mathbf{\Sigma})\mathbf{S}_t(\mathbf{I} - \eta\mathbf{\Sigma}) \tag{E.75}$$
$$+ \sigma^4 \cdot \mathbb{E}[\mathbf{\Lambda}_t \Delta_t \Delta_t^\top \mathbf{\Lambda}_t^\top] + \sigma^4\eta^2 \cdot \mathbb{E}[\mathbf{\Lambda}_t \mathbf{w}_t \mathbf{w}_t^\top \mathbf{\Lambda}_t^\top] \tag{E.76}$$
$$= (1-\sigma^2)^2 \cdot (\mathbf{I} - \eta\mathbf{\Sigma})\mathbf{S}_t(\mathbf{I} - \eta\mathbf{\Sigma}) \tag{E.77}$$
$$+ \sigma^4\big(\mathrm{Tr}(\mathbf{S}_t)\mathbf{I} + \mathrm{diag}(\mathbf{S}_t)\big) + \sigma^4 \cdot \big(\mathrm{Tr}(\mathbf{g}_t\mathbf{g}_t^\top)\mathbf{I} + \mathrm{diag}(\mathbf{g}_t\mathbf{g}_t^\top)\big). \tag{E.78}$$

Here the second identity follows from Lemma E.7 and $\mathbf{g}_t = \mathbf{w}_t + \eta\mathbf{X}^\top\mathbf{y}$. The structure of this iteration has two folds. The first part is that the gradient step, together with the average effect of the noise term, help to decay the second moment of the fluctuation. The second part is that the noise term re-allocate the fluctuation in the last step to the current step in an isotropic manner. Since $\mathrm{Tr}(\boldsymbol{A})$ prevails over $\mathrm{diag}(\boldsymbol{A})$, we can continue as

$$\mathrm{Tr}(\mathbf{S}_{t+1}) \le (1-\sigma^2)^2 \cdot \|\mathbf{I} - \eta\mathbf{\Sigma}\|_2^2 \mathrm{Tr}(\mathbf{S}_t) + \sigma^4(d+1) \cdot \big(\mathrm{Tr}(\mathbf{S}_t) + \mathrm{Tr}(\mathbf{g}_t\mathbf{g}_t^\top)\big) \tag{E.79}$$
$$\le \Big((1-\sigma^2)^2 \cdot \|\mathbf{I} - \eta\mathbf{\Sigma}\|_2^2 + \sigma^4(d+1)\Big) \cdot \mathrm{Tr}(\mathbf{S}_t) + \sigma^4(d+1)\max_{t\ge 0}\|\mathbf{g}_t\|_2^2. \tag{E.80}$$

Based on our assumption that $\sigma^2 < (d+1)^{-1}$, it holds by the convexity of the quadratic function that

$$(1-\sigma^2)^2 \cdot \|\mathbf{I} - \eta\mathbf{\Sigma}\|_2^2 + \sigma^4(d+1) \le (1-\sigma^2)^2 + \sigma^4(d+1) \tag{E.81}$$
$$\le 1 - \frac{d\sigma^2}{d+1}. \tag{E.82}$$

Plugging this back to (E.80), we have that

$$\mathrm{Tr}(\mathbf{S}_{t+1}) \le \frac{\sigma^4 \cdot (d+1) \cdot \max_{t\ge 0}\|\mathbf{g}_t\|_2^2}{1 - (1-\sigma^2)^2 \cdot \|\mathbf{I} - \eta\mathbf{\Sigma}\|_2^2 - \sigma^4(d+1)} \tag{E.83}$$
$$\le \frac{(d+1)^2\sigma^2}{d} \cdot \max_{t\ge 0}\|\mathbf{g}_t\|_2^2. \tag{E.84}$$

Now we can leverage (E.72) and get that

$$\mathbb{E}_{\boldsymbol{\epsilon},\mathbf{w}^\star,\boldsymbol{\xi}}\Big[\Big\|\frac{1}{N}\sum_{j=1}^N \Delta^{(j)}\Big\|_{\mathbf{H}}^2\Big] \le \mathbb{E}_{\boldsymbol{\epsilon},\mathbf{w}^\star}\big[N^{-1}\mathrm{Tr}(\mathbf{S}_t) \cdot \|\mathbf{H}\|_2\big] \tag{E.85}$$
$$\lesssim \frac{(d+1)^2\sigma^2}{Nd} \cdot \Big(\frac{\eta\sigma_\epsilon^2}{n} \cdot \sigma^{-4}\mathrm{Tr}(\mathbf{\Sigma}) + \mathbb{E}_{\mathbf{w}^*}[\|\mathbf{w}^*\|_2^2]\Big) \cdot \|\mathbf{H}\|_2 \tag{E.86}$$
$$\lesssim \frac{(\eta\sigma^{-2}\sigma_\epsilon^2 d \cdot \mathrm{Tr}(\mathbf{H})/n + \omega^2) \cdot \|\mathbf{H}\|_2}{N}. \tag{E.87}$$

The last inequality use that $\mathrm{Tr}(\mathbf{\Sigma}) \lesssim \mathrm{Tr}(\mathbf{H})$ with high probability. This concludes the proof for the fluctuation error. □

Now with the above lemmas, we are ready to conclude and prove Theorem D.2 for Example 2.5.

Proof of Theorem D.2 for Example 2.5. Combining Lemma E.2, Lemma E.4, Lemma E.5, and Lemma E.6 gives the desired result. □

E.4 Proof of Theorem 3.2

E.4.1 Proof for Example 2.4

Proof of Theorem 3.2 for Example 2.4. This follows directly from Theorem D.2 for Example 2.4 and the proof of Proposition 3.1. □

E.4.2 Proof for Example 2.5

Proof of Theorem 3.2 for Example 2.5. This follows from Theorem D.2 for Example 2.5, and repeating the proof of Proposition 3.2 for k^*_{Bias} and k^*_{Var} in Theorem D.2. □

E.5 Technical Results

Lemma E.7. *For any deterministic matrix* $\mathbf{A} \in \mathbb{R}^{d\times d}$ *and* $\boldsymbol{\xi} \sim \mathcal{N}(\mathbf{0}_d, \mathbf{I}_d)$*, it holds that*

$$\mathbb{E}[(\mathbf{I} - \boldsymbol{\xi}\boldsymbol{\xi}^\top)\mathbf{A}(\mathbf{I} - \boldsymbol{\xi}\boldsymbol{\xi}^\top)] = \mathrm{Tr}(\mathbf{A})\mathbf{I}_d + \mathrm{diag}(\mathbf{A}), \tag{E.88}$$

where $(\mathrm{diag}(\mathbf{A}))_{ij} = \delta_{ij} \cdot A_{ij}$ *and* δ_{ij} *is the Kronecker delta.*

Proof of Lemma E.7. Note that the (i, j)-entry of $\mathbf{I} - \boldsymbol{\xi}\boldsymbol{\xi}^\top$ is $\delta_{ij} - \xi_i\xi_j$. First of all, it is clear that whenever $|\{i, j\} \setminus \{k, l\}| \geq 1$ or $|\{k, l\} \setminus \{i, j\}| \leq 1$, we have that $\mathbb{E}[(\delta_{ij} - \xi_i\xi_j) \cdot (\delta_{kl} - \xi_k\xi_l)] = 0$. So the only non-trivial cases are that: (i) $i = j = k = l$; (ii) $\{i, j\} = \{k, l\}$ and $i \neq j$. For the first case, we have that $\mathbb{E}[(\delta_{ij} - \xi_i\xi_j) \cdot (\delta_{kl} - \xi_k\xi_l)] = \mathbb{E}[(\xi_i\xi_j)^2] = 1$. For the second case, we have that $\mathbb{E}[(1 - \xi_i^2)^2] = \mathbb{E}[\xi_i^4] - \mathbb{E}[\xi_i^2]^2 = 2$.

Given this we have for $i \neq j$ that

$$\mathbb{E}[\boldsymbol{\Lambda}\mathbf{A}\boldsymbol{\Lambda}]_{i,j} = \mathbb{E}[\sum_{k,l=1}^{d} \boldsymbol{\Lambda}_{ik}\mathbf{A}_{kl}\boldsymbol{\Lambda}_{lj}] = 0, \tag{E.89}$$

because each summand is zero since $i \neq j$. For the diagonal terms, we have that

$$\begin{aligned}
\mathbb{E}[\boldsymbol{\Lambda}\mathbf{A}\boldsymbol{\Lambda}]_{i,i} &= \mathbb{E}[\sum_{k,l=1}^{d} \boldsymbol{\Lambda}_{ik}\mathbf{A}_{kl}\boldsymbol{\Lambda}_{li}] && \text{(E.90)}\\
&= \mathbb{E}[\sum_{k=1}^{d} \boldsymbol{\Lambda}_{ik}\mathbf{A}_{kk}\boldsymbol{\Lambda}_{ki}] && \text{(E.91)}\\
&= \mathbb{E}[\sum_{k\neq i} \boldsymbol{\Lambda}_{ik}\mathbf{A}_{kk}\boldsymbol{\Lambda}_{ki}] + \mathbb{E}[\boldsymbol{\Lambda}_{ii}\mathbf{A}_{ii}\boldsymbol{\Lambda}_{ii}] && \text{(E.92)}\\
&= \mathrm{Tr}(\mathbf{A}) + \mathbf{A}_{ii}. && \text{(E.93)}
\end{aligned}$$

Thus the desired result follows. ☐

F Proofs for Section 4

Notation We let $[n]$ denote the set of indices from 1 to n. Boldface uppercase letters such as $\mathbf{X}$ represent matrices, while boldface lowercase letters such as $\mathbf{x}$ denote vectors. Specifically, $\mathbf{x}[i]$ denotes the i-th element of $\mathbf{x}$.

F.1 Proof of Theorem 4.1

Proof of Proposition 4.1. Considering that we sample N different $\mathbf{w}_t$ from the distribution $\{p(\mathbf{w}_t = \mathbf{w})\}_{\mathbf{w}\in\mathcal{W}}$ to obtain $\mathbf{W} = \{\mathbf{w}_t^{(1)}, \ldots, \mathbf{w}_t^{(N)}\}$. Let $\texttt{Count}(\mathbf{w})$ represent the frequency of occurrence of $\mathbf{w}$ in $\mathbf{W}$. For each $\mathbf{w}' \in \mathcal{W} \setminus \{\mathbf{w}^*\}$, we upper bound the probability of $\texttt{Count}(\mathbf{w}') > \texttt{Count}(\mathbf{w}^*)$. To this end, we define N random variables $a_1, \cdots, a_N$ such that $a_i = 1$ if $\mathbf{w}_t^{(i)} = \mathbf{w}^*$, $a_i = -1$ if $\mathbf{w}_t^{(i)} = \mathbf{w}'$, and $a_i = 0$ otherwise. This leads to the following bound,

$$\mathbb{P}(\texttt{Count}(\mathbf{w}') > \texttt{Count}(\mathbf{w}^*) \mid \mathbf{w}_0, \mathcal{D}) \leq \mathbb{P}\left(\sum_{i=1}^{N} a_i \leq 0 \mid \mathbf{w}_0, \mathcal{D}\right) \leq \exp\left(-\left(p(\mathbf{w}_t = \mathbf{w}^*) - p(\mathbf{w}_t = \mathbf{w}')\right)^2 \cdot \frac{N}{2}\right),$$

where the last inequality is due to Hoeffding's inequality. Then

$$\begin{aligned}
\sum_{\mathbf{w}'\in\mathcal{W}\setminus\{\mathbf{w}^*\}} \mathbb{P}(\mathbf{w}_{t,N}^{\texttt{mv}} = \mathbf{w}' \mid \mathbf{w}_0, \mathcal{D}) &\leq \sum_{\mathbf{w}'\in\mathcal{W}\setminus\{\mathbf{w}^*\}} \mathbb{P}(\texttt{Count}(\mathbf{w}') > \texttt{Count}(\mathbf{w}^*) \mid \mathbf{w}_0, \mathcal{D})\\
&\leq \sum_{\mathbf{w}'\in\mathcal{W}\setminus\{\mathbf{w}^*\}} \exp\left(-\frac{N}{2} \cdot \left(p(\mathbf{w}^*) - p(\mathbf{w}')\right)^2\right)\\
&\leq |\mathcal{W} \setminus \{\mathbf{w}^*\}| \cdot \exp\left(-\frac{N}{2} \cdot \Delta_t^2\right),
\end{aligned}$$

where the final inequality is based on the definition of $\Delta_t = p(\mathbf{w}^*) - \max_{\mathbf{w}' \in \mathcal{W} \setminus \{\mathbf{w}^*\}} p(\mathbf{w}')$. Consequently,

$$\mathbb{P}(\mathbf{w}_{t,N}^{\mathtt{mv}} = \mathbf{w}^* \mid \mathbf{w}_0, \mathcal{D}) \geq 1 - \sum_{\mathbf{w}' \in \mathcal{W} \setminus \{\mathbf{w}^*\}} \mathbb{P}(\mathbf{w}_{t,N}^{\mathtt{mv}} = \mathbf{w}' \mid \mathbf{w}_0, \mathcal{D}) \geq 1 - |\mathcal{W}| \cdot \exp\left(-\frac{N}{2} \cdot \Delta_t^2\right).$$

This completes the proof of Proposition 4.1. □

F.2 Proof of Theorem 4.2

Here, we first establish bounds for each element in $\tilde{\mathbf{w}}_t$ in Theorem F.1. Next, in Theorem F.2, we prove $\mathbf{w}_T$ will converge to $\mathbf{w}^*$ for both greedy decoding and majority vote algorithm. Lastly, in Theorem F.3, we demonstrate the convergence rate for greedy decoding as shown in Theorem 4.2.

Lemma F.1. *Given* $\tilde{\mathbf{w}}_t = \mathbf{w}_{t-1} - \frac{1}{n}\left(\mathbf{X}\mathbf{X}^\top \mathbf{w}_{t-1} - \mathbf{X}\mathbf{Y}^\top\right)$, *where* $\mathbf{Y} = \mathbf{w}^*\mathbf{X} + \epsilon$, *We define* $\mathcal{E}_1$ *as follows:*

$$\mathcal{E}_1 := \left\{\begin{array}{l} \mathbf{w}^*[i] + \dfrac{2k + \sigma_\epsilon}{n^{1/4}} \geq \tilde{\mathbf{w}}_t[i] \geq \mathbf{w}^*[i] - \dfrac{2k + \sigma_\epsilon}{n^{1/4}}, \\ \textit{specifically when } \mathbf{w}_{t-1} = \mathbf{w}^*, \mathbf{w}^*[i] + \dfrac{\sigma_\epsilon}{n^{1/4}} \geq \tilde{\mathbf{w}}_t[i] \geq \mathbf{w}^*[i] - \dfrac{\sigma_\epsilon}{n^{1/4}} \end{array}\right\},$$

then $\mathcal{E}_1$ *holds with probability at least* $1 - \delta$, *where* $\delta = 2\left(d^2 + 2d\right) e^{-cn^{1/2}}$.

Proof.

$$\begin{aligned} \tilde{\mathbf{w}}_t[i] &= \mathbf{w}_{t-1}[i] - \frac{1}{n} \sum_{j \in [n], l \in [d]} \left(x_{ji} x_{jl} \mathbf{w}_{t-1}[l] - x_{ji} x_{jl} \mathbf{w}^*[l]\right) + \frac{1}{n} \sum_{j \in [n]} x_{ji} \epsilon_i \\ &= \mathbf{w}_{t-1}[i] - \frac{1}{n} \left(\mathbf{w}_{t-1}[i] - \mathbf{w}^*[i]\right) \underbrace{\sum_{j \in [n]} x_{ji}^2}_{A_i} - \frac{1}{n} \sum_{l \in [d], l \neq i} \left(\mathbf{w}_{t-1}[l] - \mathbf{w}^*[l]\right) \underbrace{\sum_{j \in [n]} (x_{ji} x_{jl})}_{B_{il}} + \frac{1}{n} \sum_{j \in [n]} x_{ji} \epsilon_i \\ &= \mathbf{w}_{t-1}[i] - \frac{1}{n} \left(\mathbf{w}_{t-1}[i] - \mathbf{w}^*[i]\right) A_i - \frac{1}{n} \sum_{l \in [d], l \neq i} \left(\mathbf{w}_{t-1}[l] - \mathbf{w}^*[l]\right) B_{il} + \frac{1}{n} \sum_{j \in [n]} x_{ji} \epsilon_i. \end{aligned} \tag{F.1}$$

Since $x_{ij} \sim \mathcal{N}(0, 1)$ for any i, j, by Lemma 2.7.7 and Bernstein's inequality in [45], there exists an absolute constant c_1 such that

$$\mathbb{P}\{|\sum_i x_{ji} x_{jl}| \leq t\} \leq 2 \exp\left(-c_1 \min\left(\frac{t^2}{\sum_j ||x_{ji} x_{jl}||_{\psi_i}^2}, \frac{t}{\max_j ||x_{ji} x_{jl}||_{\psi_i}}\right)\right),$$

where $||.||_{\psi_1}$ denotes to the sub-exponential norm. Besides, $||x_{ji} x_{jl}||_{\psi_i} \leq ||x_{ji}||_{\psi_2} \cdot ||x_{jk}||_{\psi_2} \leq C_1^2$, with the last inequality derived from the properties of the Gaussian distribution, where C_1 is a constant. Furthermore, we have:

$$\mathbb{P}\{|B_{il}| \leq t_1\} \leq 2 \exp\left(-c_1 \min\left(\frac{t_1^2}{n C_1^4}, \frac{t_1}{C_1^2}.\right)\right) \tag{F.2}$$

Similarly we have

$$\mathbb{P}\{|\sum_{j \in [n]} x_{ji} \epsilon_i| \leq t_2\} \leq 2 \exp\left(-c_2 \min\left(\frac{t_2^2}{n C_1^4 \sigma_\epsilon^2}, \frac{t_2}{C_1^2 \sigma_\epsilon}.\right)\right) \tag{F.3}$$

For $A_i = \sum_{j \in [n]} x_{ji}^2$, since $x_{ji}^2 - 1$ are sub-exponential and mean zero random variables, we can directly apply Bernstein's inequality to obtain:

$$\mathbb{P}\{|A_i - n| \leq t_3\} \leq 2 \exp\left(-c_3 \min\left(\frac{t_3^2}{n C_3^4}, \frac{t_3}{C_3^2}\right)\right) \tag{F.4}$$

By setting $t_1 = t_3 = n^{3/4}, t_2 = \sigma_\epsilon n^{3/4}$, $c = \frac{\min(c_1,c_2,c_3)}{\max\left(C_1^4,C_2^4,C_3^4,C_1^2,C_2^2,C_3^2\right)}$, and applying the derived Equation F.2, Equation F.3, Equation F.4 for all $i, l \in [d]$, we establish that

$$\begin{aligned} &|B_{il}| \leq n^{3/4} && \forall i,l \in [d]; \\ &|\sum_{j\in[n]} x_{ji}\epsilon_i| \leq \sigma_\epsilon n^{3/4} && \forall i \in [d]; \\ &|A_i - n| \leq n^{3/4} && \forall i \in [d], \end{aligned} \tag{F.5}$$

holds with a probability of at least $1 - 2\left(d^2 + 2d\right) e^{-cn^{1/2}}$. Hereafter, we condition on Equation F.5. By combining Equation F.5 with Equation F.1, the following equation is obtained:

$$\begin{aligned} \tilde{\mathbf{w}}_t[i] &= \mathbf{w}_{t-1}[i] - \frac{1}{n}\left(\mathbf{w}_{t-1}[i] - \mathbf{w}^*[i]\right) A_i - \frac{1}{n} \sum_{l\in[d], l\neq i} \left(\mathbf{w}_{t-1}[l] - \mathbf{w}^*[l]\right) B_{il} + \frac{1}{n}\sum_{j\in[n]} x_{ji}\epsilon_i \\ &\leq \mathbf{w}^*[i] + \frac{1}{n^{1/4}} \sum_{l\in[d]} |\mathbf{w}_{t-1}[l] - \mathbf{w}^*[l]| + \frac{\sigma_\epsilon}{n^{1/4}} \\ &\leq \mathbf{w}^*[i] + \frac{2k+\sigma_\epsilon}{n^{1/4}}, \end{aligned}$$

the final inequality is by $||\mathbf{w}_t||_0 = k\ (t \geq 1)$ and $||\mathbf{w}_0||_0 = 0$. Similarly we have.

$$\begin{aligned} \tilde{\mathbf{w}}_t[i] &\geq \mathbf{w}^*[i] - \frac{1}{n^{1/4}} \sum_{l\in[d]} |\mathbf{w}_{t-1}[l] - \mathbf{w}^*[l]| - \frac{\sigma_\epsilon}{n^{1/4}} \\ &\geq \mathbf{w}^*[i] - \frac{2k+\sigma_\epsilon}{n^{1/4}} \end{aligned}$$

Specifically, when $\mathbf{w}_{t-1} = \mathbf{w}^*$,

$$\mathbf{w}^*[i] + \frac{\sigma_\epsilon}{n^{1/4}} \geq \tilde{\mathbf{w}}_t[i] \geq \mathbf{w}^*[i] - \frac{\sigma_\epsilon}{n^{1/4}}.$$

□

Without loss of generality, in the following we assume the first k elements of $\mathbf{w}^*$ are 1, and others are 0. We define $\mathcal{C}^{(m)}$ as the set of all possible permutations for $[m]$.

Lemma F.2 (Perfect Accuracy for Both Greedy Decoding and Majority Vote)**.** *Given* $\tilde{\mathbf{w}}_t = \mathbf{w}_{t-1} - \frac{1}{n}\left(\mathbf{X}\mathbf{X}^\top \mathbf{w}_{t-1} - \mathbf{X}\mathbf{Y}^\top\right)$, *where* $\mathbf{Y} = \mathbf{w}^*\mathbf{X} + \epsilon$, *suppose* $\mathcal{E}_1$ *holds,* $\frac{2k+\sigma_\epsilon}{n^{1/4}} < \frac{1}{3}$ *and sampling number* N *is sufficient large, then for all* $t \geq 1$, *we have*

$$\mathbf{w}_t^{\mathtt{maj}\cdot N} = \mathbf{w}_t^{\mathtt{greedy}} = \mathbf{w}^*.$$

Proof. Given that $\mathcal{E}_1$ holds, for $t \geq 1$:

$$\begin{cases} \tilde{\mathbf{w}}_t[i] \geq 1 - \frac{2k+\sigma_\epsilon}{n^{1/4}} > 1/2 & i \leq k \\ \tilde{\mathbf{w}}_t[i] \leq \frac{2k+\sigma_\epsilon}{n^{1/4}} < 1/2 & k < i \leq d \end{cases}.$$

In this case we observe that $\tilde{\mathbf{w}}_t[i] > \tilde{\mathbf{w}}_t[j]$ for all $i \leq k$ and $k < i \leq d$. Without loss of generality, we further assume

$$\tilde{\mathbf{w}}_t[1] \geq \tilde{\mathbf{w}}_t[2] \geq \cdots \geq \tilde{\mathbf{w}}_t[k] > \tilde{\mathbf{w}}_t[k+1] \geq \tilde{\mathbf{w}}_t[k+2] \geq \cdots \geq \tilde{\mathbf{w}}_t[d].$$

For $p_{\tilde{\mathbf{w}}_t}[i] = \frac{\max(0,\tilde{\mathbf{w}}_t)}{\sum_{j=1}^d \max(0,\tilde{\mathbf{w}}_t)}$, we also have

$$p_{\tilde{\mathbf{w}}_t}[1] \geq p_{\tilde{\mathbf{w}}_t}[2] \geq \cdots \geq p_{\tilde{\mathbf{w}}_t}[k] > p_{\tilde{\mathbf{w}}_t}[k+1] \geq p_{\tilde{\mathbf{w}}_t}[k+2] \geq \cdots \geq p_{\tilde{\mathbf{w}}_t}[d].$$

Then for $\mathbf{w}' \in \mathcal{W}_{/\mathbf{w}^*}$ where the index of nonzero elements are $e_1, e_2, \ldots, e_k$ (in increasing order), we have

$$\begin{aligned} &\mathbb{P}\left(\mathbf{w}_t = \mathbf{w}^* | \mathbf{w}_{t-1}\right) - \mathbb{P}\left(\mathbf{w}_1 = \mathbf{w}' | \mathbf{w}_{t-1}\right) \\ &= \sum_{(i_1,\ldots,i_k)\in\mathcal{C}^{(k)}} \left(p_{\tilde{\mathbf{w}}_t}[i_1] \cdot \frac{p_{\tilde{\mathbf{w}}_t}[i_2]}{1 - p_{\tilde{\mathbf{w}}_t}[i_1]} \cdots \frac{p_{\tilde{\mathbf{w}}_t}[i_k]}{1 - \sum_{j<k} p_{\tilde{\mathbf{w}}_t}[i_j]} - p_{\tilde{\mathbf{w}}_t}[e_{i_1}] \cdot \frac{p_{\tilde{\mathbf{w}}_t}[e_{i_2}]}{1 - p_{\tilde{\mathbf{w}}_t}[e_{i_1}]} \cdots \frac{p_{\tilde{\mathbf{w}}_t}[e_{i_k}]}{1 - \sum_{j<k} p_{\tilde{\mathbf{w}}_t}[e_{i_j}]} \right) \\ &> 0, \end{aligned}$$

the last inequality holds because $p_{\tilde{\mathbf{w}}_t}[i] \geq p_{\tilde{\mathbf{w}}_t}[e_i]$ for all $i < k$ and $p_{\tilde{\mathbf{w}}_t}[k] > p_{\tilde{\mathbf{w}}_t}[e_k]$, thus for $t \geq 1$:

$$\mathbb{P}\left(\mathbf{w}_t = \mathbf{w}^*|\mathbf{w}_{t-1}\right) > \mathbb{P}\left(\mathbf{w}_t = \mathbf{w}'|\mathbf{w}_{t-1}\right) \, \forall \, \mathbf{w}' \in \mathcal{W}_{/\mathbf{w}^*}, \mathbf{w}_{t-1} \in \mathcal{W},$$

Since greedy decoding selects the $\mathbf{w}$ with highest probability, $\mathbf{w}_t^{\texttt{greedy}} = \mathbf{w}^*$ for all $t \geq 1$. Additionally,

$$\begin{aligned}
\mathbb{P}\left(\mathbf{w}_t = \mathbf{w}^*|\mathbf{w}_0\right) &= \sum_{\mathbf{w} \in \mathcal{W}} \mathbb{P}\left(\mathbf{w}_t = \mathbf{w}^*|\mathbf{w}_{t-1} = \mathbf{w}\right) \mathbb{P}\left(\mathbf{w}_{t-1} = \mathbf{w}|\mathbf{w}_0\right) \\
&> \sum_{\mathbf{w} \in \mathcal{W}} \mathbb{P}\left(\mathbf{w}_t = \mathbf{w}'|\mathbf{w}_{t-1} = \mathbf{w}\right) \mathbb{P}\left(\mathbf{w}_{t-1} = \mathbf{w}|\mathbf{w}_0\right) \\
&= \mathbb{P}\left(\mathbf{w}_t = \mathbf{w}'|\mathbf{w}_0\right).
\end{aligned}$$

This implies $\mathbb{P}\left(\mathbf{w}_t = \mathbf{w}^*|\mathbf{w}_0\right) > \mathbb{P}\left(\mathbf{w}_t = \mathbf{w}'|\mathbf{w}_0\right)$ for all $\mathbf{w} \in \mathcal{W}_{/\mathbf{w}^*}$, and according to Theorem 4.1, majority vote will choose $\mathbf{w}_{t,N}^{\texttt{mv}} = \mathbf{w}^*$ with sufficient large sampling number N. □

Lemma F.3 (Convergence Rate for Majority Vote)**.** *Given* $\tilde{\mathbf{w}}_t = \mathbf{w}_{t-1} - \frac{1}{n}\left(\mathbf{X}\mathbf{X}^\top \mathbf{w}_{t-1} - \mathbf{X}\mathbf{Y}^\top\right)$, *where* $\mathbf{Y} = \mathbf{w}^*\mathbf{X} + \epsilon$, *suppose* $\mathcal{E}_1$ *holds and* $\frac{2k+\sigma_\epsilon}{n^{1/4}} < \frac{1}{3}$ *, then*

$$\mathbb{P}\left(\mathbf{w}_t = \mathbf{w}^*|\mathbf{w}_0\right) - \max_{\mathbf{w}' \in \mathcal{W}_{/\mathbf{w}^*}} \mathbb{P}\left(\mathbf{w}_t = \mathbf{w}'|\mathbf{w}_0\right) \geq \frac{p_{\texttt{trans}}}{p_{\texttt{trans}} + 1 - p_{\texttt{recurr}}} \left(1 - \left(p_{\texttt{recurr}} - p_{\texttt{trans}}\right)^{t-1}\right).$$

Where

$$\begin{aligned}
p_{\texttt{trans}} &= \left(1 - \frac{2k + \sigma_\epsilon}{n^{1/4} - (2k + \sigma_\epsilon)}\right) \frac{1}{d^k}, \\
p_{\texttt{recurr}} &= \left(1 - \frac{\sigma_\epsilon}{n^{1/4} - \sigma_\epsilon}\right) \left(\frac{n^{1/4} - \sigma_\epsilon}{n^{1/4} - \sigma_\epsilon + d\sigma_\epsilon}\right)^k.
\end{aligned}$$

Proof. First, when $\mathbf{w}_{t-1} = \mathbf{w}^*$, we have

$$\begin{cases} \tilde{\mathbf{w}}_t[i] \geq 1 - \frac{\sigma_\epsilon}{n^{1/4}} & i \leq k \\ \tilde{\mathbf{w}}_t[i] \leq \frac{\sigma_\epsilon}{n^{1/4}} & k < i \leq d \end{cases}$$

Let $\tau = \frac{\sigma_\epsilon}{n^{1/4}}$. For $p_{\tilde{\mathbf{w}}_t}[i] = \frac{\max(0, \tilde{\mathbf{w}}_t)}{\sum_{j=1}^d \max(0, \tilde{\mathbf{w}}_t)}$ and $i \leq k$:

$$p_{\tilde{\mathbf{w}}_t}[i] \geq \frac{1 - \tau}{k\left(1 - \tau\right) + d\tau} = \frac{1}{k} \frac{k\left(1 - \tau\right)}{k\left(1 - \tau\right) + d\tau}$$

Hence,

$$\begin{aligned}
\mathbb{P}\left(\mathbf{w}_t = \mathbf{w}^*|\mathbf{w}_{t-1} = \mathbf{w}^*\right) &= \sum_{(i_1, \ldots, i_k) \in \mathcal{C}^{(k)}} \left(p_{\tilde{\mathbf{w}}_t}[i_1] \cdot \frac{p_{\tilde{\mathbf{w}}_t}[i_2]}{1 - p_{\tilde{\mathbf{w}}_t}[i_1]} \cdots \frac{p_{\tilde{\mathbf{w}}_t}[i_k]}{1 - \sum_{j<k} p_{\tilde{\mathbf{w}}_t}[i_j]}\right) \\
&\geq \frac{\left(\frac{1}{k} - \frac{d\tau}{(k(1-\tau)+d\tau)k}\right)^k k!}{\prod_{m=1}^{k-1} \left(1 - m\left(\frac{1}{k} - \frac{d\tau}{(k(1-\tau)+d\tau)k}\right)\right)} \\
&\geq \left(\frac{1 - \tau}{1 + (d-1)\tau}\right)^k
\end{aligned}$$

the last inequality is by let $v = \frac{k(1-\tau)}{k(1-\tau)+d\tau}$

$$\begin{aligned}
\frac{\left(\frac{v}{k}\right)^k k!}{\prod_{m=1}^{k-1}\left(1 - m\frac{v}{k}\right)} &\geq \frac{v^k \left(\frac{1}{k}\right)^k k!}{\prod_{m=1}^{k-1}\left(\left(k - (k-1)v\right)\left(1 - m\frac{1}{k}\right)\right)} \\
&= \frac{v^k}{\left(k - (k-1)v\right)^{k-1}} \frac{\left(\frac{1}{k}\right)^k k!}{\prod_{m=1}^{k-1}\left(1 - m\frac{1}{k}\right)} \\
&\geq \left(\frac{v}{k - (k-1)v}\right)^k = \left(\frac{1 - \tau}{1 - \tau + d\tau}\right)^k
\end{aligned}$$

Next, for $\mathbf{w}' \in \mathcal{W}_{/\mathbf{w}^*}$ where the index of nonzero elements are $e_1, e_2, \ldots, e_k$ (increasing order), we have:

$$\begin{aligned}
&\mathbb{P}\left(\mathbf{w}_t = \mathbf{w}^* | \mathbf{w}_{t-1}\right) - \mathbb{P}\left(\mathbf{w}_1 = \mathbf{w}' | \mathbf{w}_{t-1}\right) \\
&= \sum_{(i_1,\ldots,i_k)\in\mathcal{C}^{(k)}} \left(p_{\tilde{\mathbf{w}}_t}[i_1] \cdot \frac{p_{\tilde{\mathbf{w}}_t}[i_2]}{1 - p_{\tilde{\mathbf{w}}_t}[i_1]} \cdots \frac{p_{\tilde{\mathbf{w}}_t}[i_k]}{1 - \sum_{j<k} p_{\tilde{\mathbf{w}}_t}[i_j]} - p_{\tilde{\mathbf{w}}_t}[e_{i_1}] \cdot \frac{p_{\tilde{\mathbf{w}}_t}[e_{i_2}]}{1 - p_{\tilde{\mathbf{w}}_t}[e_{i_1}]} \cdots \frac{p_{\tilde{\mathbf{w}}_t}[e_{i_k}]}{1 - \sum_{j<k} p_{\tilde{\mathbf{w}}_t}[e_{i_j}]} \right) \\
&> \left(\prod_{i=1}^{k} p_{\tilde{\mathbf{w}}_t}[i] - \prod_{i=1}^{k} p_{\tilde{\mathbf{w}}_t}[e_i] \right) \sum_{(i_1,\ldots,i_k)\in\mathcal{C}^{(k)}} \left(\frac{1}{1 - p_{\tilde{\mathbf{w}}_t}[i_1]} \cdots \frac{1}{1 - \sum_{j<k} p_{\tilde{\mathbf{w}}_t}[i_j]} \right) \\
&> \left(1 - \frac{p_{\tilde{\mathbf{w}}_t}[e_i]}{p_{\tilde{\mathbf{w}}_t}[i]} \right) \prod_{i=1}^{k} p_{\tilde{\mathbf{w}}_t}[i] \sum_{(i_1,\ldots,i_k)\in\mathcal{C}^{(k)}} \left(\frac{1}{1 - p_{\tilde{\mathbf{w}}_t}[i_1]} \cdots \frac{1}{1 - \sum_{j<k} p_{\tilde{\mathbf{w}}_t}[i_j]} \right) \\
&= \left(1 - \frac{p_{\tilde{\mathbf{w}}_t}[e_i]}{p_{\tilde{\mathbf{w}}_t}[i]} \right) \mathbb{P}\left(\mathbf{w}_t = \mathbf{w}^* | \mathbf{w}_{t-1}\right)
\end{aligned}$$

Given that $\mathbb{P}\left(\mathbf{w}_t = \mathbf{w}^* | \mathbf{w}_{t-1}\right) > \mathbb{P}\left(\mathbf{w}_t = \mathbf{w}' | \mathbf{w}_{t-1}\right)$ for $\mathbf{w}' \in \mathcal{W}_{/\mathbf{w}^*}$, we have $\mathbb{P}\left(\mathbf{w}_t = \mathbf{w}^* | \mathbf{w}_{t-1}\right) > \frac{1}{|\mathcal{W}|} \geq \frac{1}{d^k}$, when $\mathcal{E}_1$ holds:

$$\mathbb{P}\left(\mathbf{w}_t = \mathbf{w}^* | \mathbf{w}_{t-1}\right) - \mathbb{P}\left(\mathbf{w}_1 = \mathbf{w}' | \mathbf{w}_{t-1}\right) > \left(1 - \frac{\frac{2k+\sigma_\epsilon}{n^{1/4}}}{1 - \frac{2k+\sigma_\epsilon}{n^{1/4}}} \right) \frac{1}{d^k}$$

Specifically,

$$\mathbb{P}\left(\mathbf{w}_t = \mathbf{w}^* | \mathbf{w}^*\right) - \mathbb{P}\left(\mathbf{w}_1 = \mathbf{w}' | \mathbf{w}^*\right) > \left(1 - \frac{\frac{\sigma_\epsilon}{n^{1/4}}}{1 - \frac{\sigma_\epsilon}{n^{1/4}}} \right) \mathbb{P}\left(\mathbf{w}_t = \mathbf{w}^* | \mathbf{w}^*\right)$$

Therefore,

$$\begin{aligned}
&\mathbb{P}\left(\mathbf{w}_t = \mathbf{w}^* | \mathbf{w}_0\right) - \mathbb{P}\left(\mathbf{w}_t = \mathbf{w}' | \mathbf{w}_0\right) \\
&= \sum_{\mathbf{w}\in\mathcal{W}} \left(\mathbb{P}\left(\mathbf{w}_t = \mathbf{w}^* | \mathbf{w}_{t-1} = \mathbf{w}\right) - \mathbb{P}\left(\mathbf{w}_t = \mathbf{w}' | \mathbf{w}_{t-1} = \mathbf{w}\right) \right) \mathbb{P}\left(\mathbf{w}_{t-1} = \mathbf{w} | \mathbf{w}_0\right) \\
&> \sum_{\mathbf{w}\in\mathcal{W}_{/\mathbf{w}^*}} \left(1 - \frac{2k+\sigma_\epsilon}{n^{1/4} - (2k+\sigma_\epsilon)} \right) \mathbb{P}\left(\mathbf{w}_t = \mathbf{w}^* | \mathbf{w}_{t-1} = \mathbf{w}\right) \mathbb{P}\left(\mathbf{w}_{t-1} = \mathbf{w} | \mathbf{w}_0\right) \\
&\quad + \left(1 - \frac{\sigma_\epsilon}{n^{1/4} - \sigma_\epsilon} \right) \mathbb{P}\left(\mathbf{w}_t = \mathbf{w}^* | \mathbf{w}^*\right) \mathbb{P}\left(\mathbf{w}_{t-1} = \mathbf{w}^* | \mathbf{w}_0\right) \\
&> \left(1 - \frac{2k+\sigma_\epsilon}{n^{1/4} - (2k+\sigma_\epsilon)} \right) \frac{1}{d^k} \sum_{\mathbf{w}\in\mathcal{W}_{/\mathbf{w}^*}} \mathbb{P}\left(\mathbf{w}_{t-1} = \mathbf{w} | \mathbf{w}_0\right) + \left(1 - \frac{\sigma_\epsilon}{n^{1/4} - \sigma_\epsilon} \right) \left(\frac{1-\tau}{1-\tau+d\tau} \right)^k \mathbb{P}\left(\mathbf{w}_{t-1} = \mathbf{w}^* | \mathbf{w}_0\right) \\
&= \underbrace{\left(1 - \frac{2k+\sigma_\epsilon}{n^{1/4} - (2k+\sigma_\epsilon)} \right) \frac{1}{d^k}}_{p_{\texttt{trans}}} \left(1 - \mathbb{P}\left(\mathbf{w}_{t-1} = \mathbf{w}^* | \mathbf{w}_0\right) \right) + \underbrace{\left(1 - \frac{\sigma_\epsilon}{n^{1/4} - \sigma_\epsilon} \right) \left(\frac{n^{1/4} - \sigma_\epsilon}{n^{1/4} - \sigma_\epsilon + d\sigma_\epsilon} \right)^k}_{p_{\texttt{recurr}}} \mathbb{P}\left(\mathbf{w}_{t-1} = \mathbf{w}^* | \mathbf{w}_0\right) \\
&> \left(p_{\texttt{recurr}} - p_{\texttt{trans}} \right)^{t-1} \left(\mathbb{P}\left(\mathbf{w}_1 = \mathbf{w}^* | \mathbf{w}_0\right) - \frac{p_{\texttt{trans}}}{p_{\texttt{trans}} + 1 - p_{\texttt{recurr}}} \right) + \frac{p_{\texttt{trans}}}{p_{\texttt{trans}} + 1 - p_{\texttt{recurr}}} \\
&> \frac{p_{\texttt{trans}}}{p_{\texttt{trans}} + 1 - p_{\texttt{recurr}}} \left(1 - \left(p_{\texttt{recurr}} - p_{\texttt{trans}} \right)^{t-1} \right)
\end{aligned}$$

□

F.3 Proof of Theorem 4.3

To prove Theorem 4.3, we first demonstrate that the majority vote algorithm can achieve perfect accuracy with a high probability given a sufficient large sampling number N (by combining Theorem F.4 and Theorem F.5). Subsequently, for the greedy decoding algorithm, we prove that with high probability, $\mathbf{w}_t^{\texttt{greedy}}$ will transition between states $\mathbf{w}'$ and $\mathbf{w}''$, where $\mathbf{w}', \mathbf{w}'' \neq \mathbf{w}^*$.

In the following, as we consider the case where $k=1$, we define $\mathbb{1}_i = [0, \ldots, \underset{\underset{i\text{-th}}{\downarrow}}{1}, 0, \ldots]$ be a vector with a value of 1 at the i-th element and 0 elsewhere. Without loss of generality, we assume $\mathbf{w}^* = \mathbb{1}_1$.

Lemma F.4. *Consider the case where $n = k = 1, \sigma_\epsilon = 0$, and denote the in-context example as $\left(\mathbf{x}, \mathbf{w}^\top \mathbf{x}\right)$. Then:*

$$\mathbb{P}\left(\mathbf{w}_{t+2} = \mathbf{w}^* | \mathbf{w}_t = \mathbf{w}\right) > 0$$

Holds for all $\mathbf{w} \in \mathcal{W}$ with probability at least $1 - \frac{1}{2^{d-1}}$.

Proof.

$$\mathbb{P}\left(\mathbf{w}_{t+2} = \mathbf{w}^* | \mathbf{w}_t = \mathbf{w}\right) = \sum_{\mathbf{w}' \in \mathcal{W}} \mathbb{P}\left(\mathbf{w}_{t+2} = \mathbf{w}^* | \mathbf{w}_{t-1} = \mathbf{w}'\right) \mathbb{P}\left(\mathbf{w}_{t+1} = \mathbf{w}' | \mathbf{w}_t = \mathbf{w}\right)$$

It suffices to demonstrate the existence of a $\mathbf{w}' \in \mathcal{W}$, such that $\mathbb{P}\left(\mathbf{w}_{t+2} = \mathbf{w}^* | \mathbf{w}_{t-1} = \mathbf{w}'\right) \mathbb{P}\left(\mathbf{w}_{t+1} = \mathbf{w}' | \mathbf{w}_t = \mathbf{w}\right) > 0$.

Without losing generality, we let $x_1 > 0$, $\mathbf{w}_t = \mathbb{1}_l$ and for $\mathbf{x} = [x_1, x_2, \ldots, x_d]$ we let $x_1 > 0$, $x_2 \geq x_3 \cdots \geq x_d$. We have:

$$\tilde{\mathbf{w}}_{t+1}[i] = \mathbf{w}_t[i] - \sum_{j \in [d]} \left(x_i x_j \left(\mathbf{w}_{t-1}[j] - \mathbf{w}^*[j]\right)\right)$$

$$\begin{cases} \tilde{\mathbf{w}}_{t+1}[i] = x_i \left(x_1 - x_l\right) & \text{if } i \neq l \\ \tilde{\mathbf{w}}_{t+1}[i] = 1 + x_l \left(x_1 - x_l\right) & \text{if } i = l \end{cases}.$$

If $x_1 - x_l > 0$, then $\tilde{\mathbf{w}}_{t+1}[1] > 0$, implying the existence of $\mathbf{w}' = \mathbf{w}^*$, such that:

$$\begin{aligned} &\mathbb{P}\left(\mathbf{w}_{t+2} = \mathbf{w}^* | \mathbf{w}_{t-1} = \mathbf{w}'\right) \mathbb{P}\left(\mathbf{w}_{t+1} = \mathbf{w}' | \mathbf{w}_t = \mathbf{w}\right) \\ =&\mathbb{P}\left(\mathbf{w}_{t+2} = \mathbf{w}^* | \mathbf{w}_{t-1} = \mathbf{w}*\right) \mathbb{P}\left(\mathbf{w}_{t+1} = \mathbf{w}* | \mathbf{w}_t = \mathbf{w}\right) \\ =&\frac{x_1 \left(x_1 - x_l\right)}{\sum_{i \in [d]} \max\left(0, \tilde{\mathbf{w}}_{t+1}[i]\right)} > 0 \end{aligned}$$

If $x_1 - x_l < 0$, we consider the case where $x_d < 0$, which occurs with a probability of at least $1 - \frac{1}{2^{d-1}}$. In this case, we ensure $x_d < 0$ to satisfy $x_d \left(x_1 - x_l\right) > 0$. Subsequently, leveraging the condition $x_1 - x_d > 0$, we can choose $\mathbf{w}' = \mathbb{1}_d$ such that:

$$\begin{aligned} &\mathbb{P}\left(\mathbf{w}_{t+2} = \mathbf{w}^* | \mathbf{w}_{t-1} = \mathbf{w}'\right) \mathbb{P}\left(\mathbf{w}_{t+1} = \mathbf{w}' | \mathbf{w}_t = \mathbf{w}\right) \\ \geq&\frac{x_d \left(x_1 - x_l\right)}{\sum_{i \in [d]} \max\left(0, \tilde{\mathbf{w}}_{t+1}[i]\right)} \cdot \frac{x_1 \left(x_1 - x_d\right)}{\sum_{i \in [d]} \max\left(0, \tilde{\mathbf{w}}_{t+2}[i]\right)} > 0 \end{aligned}$$

□

Lemma F.5. *Consider the case where $n = k = 1, \sigma_\epsilon = 0$, and denote the in-context example as $\left(\mathbf{x}, \mathbf{w}^\top \mathbf{x}\right)$.There exists a $\zeta > 0$ such that for reasoning steps $T > \frac{2 \ln 1/2}{\ln 1 - \zeta}$ and sufficient large sampling number N, it holds that*

$$\mathbf{w}_{T,N}^{\mathtt{mv}} = \mathbf{w}^*,$$

with probability at least $1 - \frac{1}{2^{d-1}}$.

Proof. Referring to Theorem F.4, with probability at least $1 - \frac{1}{2^{d-1}}$, $\mathbb{P}\left(\mathbf{w}_{t+2} = \mathbf{w}^* | \mathbf{w}_t = \mathbf{w}\right) > 0$ holds for all $\mathbf{w} \in \mathcal{W}$, define

$$\zeta = \min_{\mathbf{w} \in \mathcal{W}} \mathbb{P}\left(\mathbf{w}_{t+2} = \mathbf{w}^* | \mathbf{w}_t = \mathbf{w}\right).$$

Assume $t = 2q+1$ (if not, since $\mathbb{P}(\mathbf{w}_t = \mathbf{w}^*|\mathbf{w}_0) \geq \mathbb{P}(\mathbf{w}_{t-1} = \mathbf{w}^*|\mathbf{w}_0)$, we can set $t-1 = 2q+1$)

$$\begin{aligned}
&\mathbb{P}(\mathbf{w}_{2q+1} = \mathbf{w}^*|\mathbf{w}_0) \\
=&\sum_{\mathbf{w}\in\mathcal{W}} \mathbb{P}(\mathbf{w}_{2q+1} = \mathbf{w}^*|\mathbf{w}_{2q-1} = \mathbf{w})\,\mathbb{P}(\mathbf{w}_{2q-1} = \mathbf{w}|\mathbf{w}_0) \\
=&\sum_{\mathbf{w}\in\mathcal{W}_{/\mathbf{w}^*}} \mathbb{P}(\mathbf{w}_{2q+1} = \mathbf{w}^*|\mathbf{w}_{2q-1} = \mathbf{w})\,\mathbb{P}(\mathbf{w}_{2q-1} = \mathbf{w}|\mathbf{w}_0) + \mathbb{P}(\mathbf{w}_{2q+1} = \mathbf{w}^*|\mathbf{w}_{2q-1} = \mathbf{w}^*)\,\mathbb{P}(\mathbf{w}_{2q-1} = \mathbf{w}^*|\mathbf{w}_0) \\
\geq&\zeta\left(1 - \mathbb{P}(\mathbf{w}_{2q-1} = \mathbf{w}^*|\mathbf{w}_0)\right) + \mathbb{P}(\mathbf{w}_{2q-1} = \mathbf{w}^*|\mathbf{w}_0) \\
\geq&(1-\zeta)^k\left(\mathbb{P}(\mathbf{w}_1 = \mathbf{w}^*|\mathbf{w}_0) - 1\right) + 1 \geq 1 - (1-\zeta)^k
\end{aligned}$$

If $k > \frac{\ln 1/2}{\ln(1-\zeta)}$, then $\mathbb{P}(\mathbf{w}_{2q+1} = \mathbf{w}^*|\mathbf{w}_0) > 1/2$, and therefore:

$$\mathbb{P}(\mathbf{w}_t = \mathbf{w}^*|\mathbf{w}_0) > \frac{1}{2} > 1 - \mathbb{P}(\mathbf{w}_t = \mathbf{w}^*|\mathbf{w}_0) > \mathbb{P}(\mathbf{w}_t = \mathbf{w}'|\mathbf{w}_0)\ \forall \mathbf{w}' \in \mathcal{W}_{/\mathbf{w}^*}$$

In this case, by Theorem 4.1, with sufficient large sample number N, $\mathbf{w}^{\mathtt{mv}}_{T,N} = \mathbf{w}^*$. ☐

Lemma F.6. *Consider the case where $n = k = 1, \sigma_\epsilon = 0$, and denote the in-context example as $\left(\mathbf{x}, \mathbf{w}^\top\mathbf{x}\right)$. Then*

$$\mathbf{w}^{\mathtt{greedy}}_t \neq \mathbf{w}^*$$

holds with probability at least $1 - \frac{2}{d} - \frac{1}{2^{d-1}}$.

Proof. Here, we directly construct a case where, with a high probability, the greedy decoding will become stuck between two stages and fail to reach the state $\mathbf{w}^*$.

Without loss of generality, we assume $x_1 > 0$, and we select x_2 and x_3 such that $x_2 = \max_{i>1} x_i$ and $x_3 = \max_{i>1}(-x_i)$. With a probability of $1 - \sum_{r=1}^{d-1} \frac{1}{r+1}\frac{\binom{d-1}{r}}{2^{d-1}} - \frac{1}{2^{d-1}} > 1 - \frac{2}{d} - \frac{1}{2^{d-1}}$, it holds that $x_2 > x_1 > 0$ and $x_3 < 0$.

In this case,

$$\tilde{\mathbf{w}}_1[2] = x_1x_2 > x_1x_j = \tilde{\mathbf{w}}_1[j],$$

holds for all $j \in [d], j \neq 2$. Then $\mathbf{w}^{\mathtt{greedy}}_1 = \mathbf{w}' \neq \mathbf{w}^*$ where $\mathbf{w}' = \mathbb{1}_2$. Similarly,

$$\begin{cases} \tilde{\mathbf{w}}_2[i] = x_i\,(x_1 - x_2) & \text{if } i \neq 2 \\ \tilde{\mathbf{w}}_2[i] = 1 + x_i\,(x_1 - x_2) & \text{if } i = 2 \end{cases}$$

If $\arg\max_{i\in[d]} \tilde{\mathbf{w}}_2[i] = 2$, then $\mathbf{w}^{\mathtt{greedy}}_2 = \mathbf{w}'$, thus for $\mathbf{w}^{\mathtt{greedy}}_t = \mathbf{w}' \neq \mathbf{w}^*$ holds when $t \geq 1$. . If $\arg\max_{i\in[d]} \tilde{\mathbf{w}}_2[i] \neq 2$, as $x_1 - x_2 < 0$,

$$\tilde{\mathbf{w}}_2[3] = x_3\,(x_1 - x_2) > x_i\,(x_1 - x_2) = \tilde{\mathbf{w}}_2[j],$$

holds for all $j \in [d], j \neq 3$. In this case, we have $\mathbf{w}_2 = \mathbf{w}'' \neq \mathbf{w}^*$ where $\mathbf{w}'' = \mathbb{1}_3$ and for $\tilde{\mathbf{w}}_3$:

$$\begin{cases} \tilde{\mathbf{w}}_3[i] = x_i\,(x_1 - x_3) & \text{if } i \neq 3 \\ \tilde{\mathbf{w}}_3[i] = 1 + x_i\,(x_1 - x_3) & \text{if } i = 3 \end{cases}$$

Similarly, if $\arg\max_{i\in[d]} \tilde{\mathbf{w}}_3[i] = 3$, then $\mathbf{w}^{\mathtt{greedy}}_3 = \mathbf{w}''$, thus for $\mathbf{w}^{\mathtt{greedy}}_t = \mathbf{w}'' \neq \mathbf{w}^*$ holds when $t \geq 2$.

If $\arg\max_{i\in[d]} \tilde{\mathbf{w}}_2[i] \neq 2$, as $(x_1 - x_3) > 0$, we know that $\mathbf{w}^{\mathtt{greedy}}_3 = \mathbf{w}'$, then $\mathbf{w}^{\mathtt{greedy}}_4 = \mathbf{w}''$, $\mathbf{w}^{\mathtt{greedy}}_5 = \mathbf{w}'$...

In conclusion, $\mathbf{w}^{\mathtt{greedy}}_t$ will be either $\mathbf{w}'$ or $\mathbf{w}''$ for $t > 0$, thus $\mathbf{w}^{\mathtt{greedy}}_t \neq \mathbf{w}^*$ for $t > 0$. ☐

G Prompt Examples

Prompt For GSM8K with Assigned Token Budget

You are a math problem solver. I will give you a problem from the Grade School Math 8K dataset (GSM8K). At the end, provide the final answer as a single integer.
Example: Problem: There are 15 trees in the grove. Grove workers will plant trees in the grove today. After they are done, there will be 21 trees. How many trees did the grove workers plant today? Answer (You should choose different reasoning method based on different tokens limit):
Case 1 (low token budgets, for example 20): We have token limits 20. The answer is ##6##. [END]
Case 2 (medium token budgets, for example 100): We have token limits 100. 21 - 15 = 6. The answer is ##6##. [END]
Case 3 (high token budgets, for example 200): We have token limits 200. There are 15 trees originally. Then there were 21 trees after some more were planted. So there must have been 21 - 15 = 6. The answer is ##6##. [END]
Case 4 (sufficient token budgets, for example 500): We have token limits 500. There are 15 trees originally. Then there were 21 trees after some more were planted. So there must have been 21 - 15 = 6. [...(more thoughts such as check answer to satisfy tokens limit)] The answer is ##6##. [END]
Important: You should try your best to use around {token_limit} tokens in your reasoning steps.
If you feel like you are finished early, spend the extra tokens trying to double check your work until you are absolutely sure that you have the correct answer.
Here's the problem:
{problem}
Solve this problem, use around {token_limit} tokens in your reasoning, provide the final answer as a single integer, and put your final answer in this format: "The answer is ##your answer##.", and end this chat with '[END]'

For the MATH dataset, we simply replaced the "Grade School Math 8K dataset (GSM8K)" (first line in above prompt) with "MATH."

H Technical Appendices and Supplementary Material

Technical appendices with additional results, figures, graphs and proofs may be submitted with the paper submission before the full submission deadline (see above), or as a separate PDF in the ZIP file below before the supplementary material deadline. There is no page limit for the technical appendices.

NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

The checklist answers are an integral part of your paper submission. They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes] " is generally preferable to "[No] ", it is perfectly acceptable to answer "[No] " provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No] " or "[NA] " is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading "NeurIPS Paper Checklist"**,
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers**.

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims made in the abstract and introduction accurately reflect the paper's contributions and scope

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

Justification: In last section, we discuss the limitations and future works.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: For each theoretical result,we provide the full set of assumptions and a complete and correct proof, and validate them with experiments.

   Guidelines:

   - The answer NA means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: In appendix we provide our experiments details.

   Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [NA]

   Justification: we mainly evaluate open LLMs and we provide experiment settings and prompt in our paper, no need to opensource our code.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.

- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: In appendix we provide our experiments details.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: we repeat our experiments 4 times and visualize all of them in our figure.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
   - If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: the paper provide sufficient information on the computer resources.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [No]

    Justification: there is no societal impact of the work performed.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

Justification: the paper poses no such risks.

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited, the license and terms of use explicitly mentioned and properly respected.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: the paper does not release new assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: the paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA] .

Justification: the paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: We use LLMs for evaluatoin and check its test time computing performance.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.