How Do Nonlinear Transformers Acquire Generalization-Guaranteed CoT Ability?

Hongkang Li [1] **Meng Wang** [1] **Songtao Lu** [2] **Xiaodong Cui** [2] **Pin-Yu Chen** [2]

Abstract

Chain-of-Thought (CoT) is an efficient prompting method that enables the reasoning ability of large language models by augmenting the query using multiple examples with intermediate steps. Despite the empirical success, the theoretical understanding of how to train a Transformer to achieve the CoT ability remains less explored. This is primarily due to the technical challenges involved in analyzing the nonconvex optimization on nonlinear attention models. To the best of our knowledge, this work provides the first theoretical study of training Transformers with nonlinear attention to obtain the CoT generalization capability so that the resulting model can reason on unseen tasks when the input is augmented by examples of the new task. We first quantify the required training samples and iterations to train a model with CoT ability. We then prove the success of its CoT generalization on unseen tasks with distribution-shifted testing data. Moreover, we theoretically characterize the conditions for an accurate reasoning output by CoT even when the provided reasoning examples contain noises and are not always accurate. In contrast, in-context learning (ICL), which can be viewed as one-step CoT without intermediate steps, may fail to provide an accurate output when CoT does. These theoretical findings are justified through experiments.

1. Introduction

Transformer-based large-scale foundation models, such as GPT-3 (Brown et al., 2020), GPT-4 (OpenAI, 2023), LLaMa (Touvron et al., 2023a;b), and Sora (Liu et al., 2024), have demonstrated remarkable success across various tasks, including natural language processing (Brown et al., 2020; Touvron et al., 2023b), multimodal learning (OpenAI, 2023; Radford et al., 2021), and image/video generation (OpenAI, 2023; Liu et al., 2024). What is more surprising is that large language models (LLMs) demonstrate reasoning ability through the so-called "Chain-of-Thought" (CoT) method (Wei et al., 2022). The objective is to let a pre-trained LLM generate K steps of reasoning given input query $\boldsymbol{x}_{query}$ without any fine-tuning. To achieve that, the input $\boldsymbol{x}_{query}$ is augumented with l examples $\{\boldsymbol{x}_i, \{\boldsymbol{y}_{i,j}\}_{j=1}^K\}_{i=1}^l$ of a certain K-step reasoning task, where each $\boldsymbol{x}_i$ is the input with $\boldsymbol{y}_{i,j}$ as the j-th reasoning step, and $\boldsymbol{y}_{i,K}$ is the final output. A pre-trained model then takes the resulting augmented input, referred to as a *prompt*, and outputs the corresponding reasoning steps $\{\boldsymbol{z}_j\}_{j=1}^K$ for $\boldsymbol{x}_{query}$, or simply outputs $\boldsymbol{z}_K$. CoT can be viewed as an extended and more intelligent method than the previous in-context learning (ICL) method, where only input-label pairs $\{\boldsymbol{x}_i, \boldsymbol{y}_{i,K}\}_{i=1}^l$ are augmented in the prompt to predict $\boldsymbol{z}_K$ with the pre-trained model.

Inspired by the outstanding empirical performance of CoT in arithmetic reasoning (Wang et al., 2023; Zhang et al., 2023b), symbolic reasoning (Zhang et al., 2023b; Zhou et al., 2023), and commonsense reasoning (Wang et al., 2023), there have been some recent works (Li et al., 2023d; Feng et al., 2023; Li et al., 2024d; Yang et al., 2024) on the theoretical understanding of CoT. These works investigate CoT from the perspective of expressive power, i.e., they construct the Transformer architecture that is proven to have the CoT ability. They also demonstrate empirically that supervised training on pairs of CoT prompts and corresponding outputs can lead to models with CoT ability. However, none of these results theoretically address the question of why a Transformer can obtain generalization-guaranteed CoT ability by training from data with gradient-based methods. Meanwhile, another line of research (Zhang et al., 2023a; Huang et al., 2023; Wu et al., 2023; Li et al., 2024a) aims to unveil the reasons behind the ICL ability of Transformers through characterizing the training dynamics of a Transformer in the supervised setting. These analyses are specifically applicable to ICL. Therefore, a theoretical question still remains less explored, i.e.,

[1]Department of Electrical, Computer, and System Engineering, Rensselaer Polytechnic Institute, Troy, NY, USA [2]IBM Thomas J. Watson Research Center, Yorktown Heights, NY, USA. Correspondence to: Hongkang Li <lih35@rpi.edu>, Meng Wang <wangm7@rpi.edu>, Songtao Lu <songtao@ibm.com>, Xiaodong Cui <cuix@us.ibm.com>, Pin-Yu Chen <pin-yu.chen@ibm.com>.

Work presented at TF2M workshop at ICML 2024, Vienna, Austria. PMLR 235, 2024.

Why can a Transformer be trained to generalize on multi-step reasoning tasks via CoT?

1.1. Major Contributions

Following (Li et al., 2023d; Feng et al., 2023; Li et al., 2024d; Yang et al., 2024; Wen et al., 2024), we train the model in a supervised setting using prompt and label pairs. This paper provides the first theoretical analysis of the training dynamics of nonlinear Transformers to achieve CoT ability. We prove that the learned model has guaranteed CoT ability for new tasks with distribution shifts from the training tasks, even when there exist noisy and erroneous context examples in the prompt. We theoretically characterize the required number of training samples and iterations needed to train a desirable model and the number of context examples required for successful CoT reasoning with a generalization guarantee. Moreover, we provide a theoretical explanation for why CoT outperforms ICL in some cases. Our main technical contributions are as follows:

1. **A quantitative analysis of how the training can enable the CoT ability**: We theoretically analyze the training dynamics on a one-layer single-head attention-only Transformer and quantify the required number of context examples in each training sample, the total number of training samples, and the number of training iterations needed to acquire CoT ability. We illustrate that the CoT ability results from the property that the attention values of the learned model are concentrated on testing context examples with the same input patterns as the testing query during each reasoning step.

2. **A quantitative analysis of how context examples affect CoT performance**: We characterize the required number of context examples in the testing prompt for successful CoT with noise and error in contexts. Our bounds are consistent with the intuition that more accurate context examples and more similar examples to the query improve CoT accuracy.

3. **A theoretical characterization of why CoT outperforms ICL**: We provide a quantitative analysis of the requirements for successful ICL reasoning with our studied trained model. We show that successful ICL requires an additional condition that the prompt has a dominant number of correct input-label examples, while CoT does not depend on this condition. This can be viewed as one of the possible reasons why CoT outperforms ICL.

2. Problem Formulation

We study the problem of learning and generalization of K-steps reasoning tasks. Each task $f = f_K \circ \cdots f_2 \circ f_1$ is a composition of functions $\{f_i\}_{i=1}^K$ and outputs labels $\boldsymbol{z}_1, \boldsymbol{z}_2, \cdots, \boldsymbol{z}_K$ for the input $\boldsymbol{x}_{query}$. During the k-th reasoning step, $k \in [K]$, the label is $\boldsymbol{z}_k = f_k(\boldsymbol{z}_{k-1})$, where $\boldsymbol{z}_0 := \boldsymbol{x}_{query}$.

2.1. Training to acquire the Chain-of-Thought ability

Following (Li et al., 2023d; Feng et al., 2023; Li et al., 2024d; Yang et al., 2024; Wen et al., 2024), we first investigate the training on a Transformer model to obtain the CoT ability in evaluating new data and tasks. It is a supervised learning setting on pairs of prompts and labels. Different from the testing prompt that includes examples and only $\boldsymbol{x}_{query}$, the training prompt includes multiple K-steps reasoning examples and a $(k-1)$-step reasoning of $\boldsymbol{x}_{query}$ for any k in $[K]$, and the label for this prompt is $\boldsymbol{z}_k$. Specifically,

Training Prompt and Label for CoT. For every prompt and output pair from a task $f = f_K \circ \cdots f_2 \circ f_1$, we construct a prompt $\boldsymbol{P}$ that include the query input $\boldsymbol{z}_{k-1}$ by prepending l_{tr} reasoning examples and the first $k-1$ steps of the reasoning query. The prompt $\boldsymbol{P}$ of the query input $\boldsymbol{z}_{k-1}$ is formulated as:

$$\begin{aligned}
&\boldsymbol{P} = \left(\boldsymbol{E}_1, \boldsymbol{E}_2, \cdots, \boldsymbol{E}_{l_{tr}}, \boldsymbol{Q}_k\right) \in \mathbb{R}^{2d_{\mathcal{X}} \times (l_{tr}K+k)},\\
&\boldsymbol{E}_i = \begin{pmatrix} \boldsymbol{x}_i & \boldsymbol{y}_{i,1} & \cdots & \boldsymbol{y}_{i,K-1} \\ \boldsymbol{y}_{i,1} & \boldsymbol{y}_{i,2} & \cdots & \boldsymbol{y}_{i,K} \end{pmatrix},\\
&\boldsymbol{Q}_k = \begin{pmatrix} \boldsymbol{z}_0 & \boldsymbol{z}_1 & \cdots & \boldsymbol{z}_{k-2} & \boldsymbol{z}_{k-1} \\ \boldsymbol{z}_1 & \boldsymbol{z}_2 & \cdots & \boldsymbol{z}_{k-1} & \boldsymbol{0} \end{pmatrix}, i \in [l_{tr}],
\end{aligned} \tag{1}$$

where $\boldsymbol{E}_i$ is the i-th context example, and $\boldsymbol{Q}_k$ is the first k steps of the reasoning query for any k in $[K]$. We have $\boldsymbol{y}_{i,k} = f_k(\boldsymbol{y}_{i,k-1})$ and $\boldsymbol{z}_k = f_k(\boldsymbol{z}_{k-1})$ for $i \in [l_{tr}]$, $k \in [K]$ with a notation $\boldsymbol{y}_{i,0} := \boldsymbol{x}_i$. Let $\boldsymbol{p}_s$ and $\boldsymbol{p}_{query}$ be the s-th column and the last column of $\boldsymbol{P}$, respectively, for $s \in [l_{tr}K + k - 1]$. $\boldsymbol{x}_i, \boldsymbol{y}_{i,k}, \boldsymbol{z}_j \in \mathbb{R}^{d_{\mathcal{X}}}$ for $i \in [l_{tr}]$ and $j, k \in [K]$. We respectively call $\boldsymbol{x}_i$ and $\boldsymbol{y}_{i,k}$ *context* inputs and outputs of the k-th step of the ith context example. For simplicity of presentation, we denote $\boldsymbol{z}$ as the label of $\boldsymbol{P}$, which is indeed $\boldsymbol{z}_k$ for (1). All the notations are summarized in Table 1 in Appendix.

The **learning model** is a single-head, one-layer attention-only Transformer. We consider positional encoding $\{\boldsymbol{c}_k\}_{k=1}^K \in \mathbb{R}^{2d_{\mathcal{X}}}$. Following theoretical works (Jelassi et al., 2022; Huang et al., 2024), we add the positional encoding to each $\boldsymbol{p}_i$ by $\tilde{\boldsymbol{p}}_i = \boldsymbol{p}_i + \boldsymbol{c}_{(i \mod K)}$, $i \in [K(l_{tr}+1)]$. $\tilde{\boldsymbol{p}}_{query}$ is also defined by adding the corresponding $\boldsymbol{c}_k$ to $\boldsymbol{p}_{query}$. Mathematically, given a prompt $\boldsymbol{P}$ defined in (1) with len(P) (which is at most $K(l_{tr}+1)$) denoting the number of columns, it can be written as

$$\begin{aligned}
&F(\Psi; \boldsymbol{P})\\
=& \sum_{i=1}^{\text{len}(P)-1} \boldsymbol{W}_V \tilde{\boldsymbol{p}}_i \cdot \text{softmax}((\boldsymbol{W}_K \tilde{\boldsymbol{p}}_i)^\top \boldsymbol{W}_Q \tilde{\boldsymbol{p}}_{query}),
\end{aligned} \tag{2}$$

where $\boldsymbol{W}_Q, \boldsymbol{W}_K \in \mathbb{R}^{m \times (2d_{\mathcal{X}})}$, $\boldsymbol{W}_V \in \mathbb{R}^{d_{\mathcal{X}} \times (2d_{\mathcal{X}})}$ are the embedding matrices for queries, keys, and values, respectively. $\Psi := \{\boldsymbol{W}_Q, \boldsymbol{W}_K, \boldsymbol{W}_V\}$ denotes the set of all model weights. Typically, $m > 2d_{\mathcal{X}}$.

The **training problem** to enable the reasoning solves the empirical risk minimization,

$$\min_{\Psi} R_N(\Psi) := \frac{1}{N}\sum_{n=1}^{N} \ell(\Psi; \boldsymbol{P}^n, \boldsymbol{z}^n), \quad (3)$$

using N prompt and label pairs $\{\boldsymbol{P}^n, \boldsymbol{z}^n\}_{n=1}^N$. For the n-th sample, $\boldsymbol{x}_{query}^n$ and the context input $\boldsymbol{x}_i^n$ are all sampled from a distribution $\mathcal{D}$. The training task f^n is sampled from $\mathcal{T}$. k is randomly selected from $[K]$, and $\boldsymbol{P}^n$ is constructed following (1). The loss function is squared loss, i.e., $\ell(\Psi; \boldsymbol{P}^n, \boldsymbol{z}^n) = 1/2\cdot\|\boldsymbol{z}^n - F(\Psi; \boldsymbol{P}^n)\|^2$, where $F(\Psi; \boldsymbol{P}^n)$ is defined in (2).

2.2. Chain-of-Thought Inference

We then consider another K-steps reasoning task $f \in \mathcal{T}'$, whose target is to predict labels $\{\boldsymbol{z}_k\}_{k=1}^K$ given the input query $\boldsymbol{x}_{query}$. $\mathcal{T}'$ is the set of testing tasks, and $\mathcal{T}' \neq \mathcal{T}$.

Testing Prompt for CoT. The testing prompt $\boldsymbol{P}$ is composed of l_{ts} $(\leq l_{tr})$ context examples of K steps plus a query, which is constructed as

$$\begin{aligned}&\boldsymbol{P} = (\boldsymbol{E}_1, \boldsymbol{E}_2, \cdots, \boldsymbol{E}_{l_{ts}}, \boldsymbol{p}_{query}) \in \mathbb{R}^{(2d_\mathcal{X})\times(l_{ts}K+1)},\\ &\boldsymbol{p}_{query} = (\boldsymbol{x}_{query}^\top, \boldsymbol{0}^\top)^\top,\end{aligned} \quad (4)$$

where $\boldsymbol{E}_i$ follows the form in (1) for $i \in [l_{ts}]$.

We follow the CoT-I/O scheme formulated in (Li et al., 2023d; Feng et al., 2023; Li et al., 2024d; Yang et al., 2024; Park et al., 2024) as the inference method. Specifically, for a K-step CoT with l_{ts} examples on a certain $f \in \mathcal{T}'$, given the testing prompt $\boldsymbol{P}$ defined in (4), let $\boldsymbol{P}_1 = \boldsymbol{P}$ and $\boldsymbol{P}_0$ be the first $K \cdot l_{ts}$ columns of $\boldsymbol{P}$. When we use CoT prompting for prediction in the k-th step, we first generate the output $\boldsymbol{v}_k$, $k \in [K]$ via greedy decoding by feeding the k-th step prompt $\boldsymbol{P}_k$ to the trained model Ψ obtained from (3). The greedy decoding scheme means outputting the most probable token from the discrete set $\mathcal{Y}$ of all possible outputs, as stated in (5).

$$\boldsymbol{v}_k = \arg\min_{\boldsymbol{u}\in\mathcal{Y}} \frac{1}{2}\|F(\Psi; \boldsymbol{P}_k) - \boldsymbol{u}\|^2, \text{ (greedy decoding)} \quad (5)$$

Then, we use the output $\boldsymbol{v}_k$ to update $\boldsymbol{P}_k$ and use $\boldsymbol{v}_k$ as the query input to form the input prompt $\boldsymbol{P}_{k+1}$ for the next step, which is computed as

$$\begin{aligned}&\boldsymbol{P}_k = \big(\boldsymbol{P}_{k-1}\ \boldsymbol{q}_k\big) \in \mathbb{R}^{(2d_\mathcal{X})\times(Kl_{ts}+k)},\\ &\boldsymbol{P}_{k+1} = \big(\boldsymbol{P}_k\ \boldsymbol{q}_{k+1}\big) \in \mathbb{R}^{(2d_\mathcal{X})\times(Kl_{ts}+k+1)},\\ &\text{where } \boldsymbol{q}_k = \big(\boldsymbol{v}_{k-1}^\top\ \boldsymbol{v}_k^\top\big)^\top,\ \boldsymbol{q}_{k+1} = \big(\boldsymbol{v}_k^\top\ \boldsymbol{0}^\top\big)^\top,\end{aligned} \quad (6)$$

where $\boldsymbol{q}_k$ is the k-th step reasoning column for the query. The model finally outputs $\boldsymbol{v}_1, \cdots, \boldsymbol{v}_K$ as CoT result for query $\boldsymbol{x}_{query}$ by (5). The CoT process is summarized in Algorithm 2 of Appendix B.

When $K \geq 2$, following (Li et al., 2023d; Feng et al., 2023; Li et al., 2024d; Yang et al., 2024), the **CoT generalization error** given the testing query $\boldsymbol{x}_{query}$, the testing data distribution $\mathcal{D}'$, and the labels $\{\boldsymbol{z}_k\}_{k=1}^K$ on a K-steps testing task $f \in \mathcal{T}'$ is defined as

$$\begin{aligned}&\bar{R}_{CoT,\boldsymbol{x}_{query}\sim\mathcal{D}',f\in\mathcal{T}'}^f(\Psi)\\ =&\mathbb{E}_{\boldsymbol{x}_{query}\sim\mathcal{D}'}\left[\frac{1}{K}\sum_{k=1}^K \mathbb{1}[\boldsymbol{z}_k \neq \boldsymbol{v}_k]\right],\end{aligned} \quad (7)$$

which measures the average error between the output and the label of each reasoning step. A zero CoT generalization error indicates correct generations in all K steps.

2.3. In-Context Learning Inference

The ICL inference on a K-steps task $f \in \mathcal{T}'$ only predicts the final-step label by prepending examples of input and label pairs before the query. ICL can be viewed as a one-step CoT without intermediate steps. We evaluate the ICL performance with the trained model.

Testing Prompt for ICL. Mathematically, ICL is implemented by constructing $\boldsymbol{P}$ as

$$\begin{aligned}&\boldsymbol{P} = (\boldsymbol{E}_1, \cdots, \boldsymbol{E}_{l_{ts}}, \boldsymbol{p}_{query}),\\ &\text{where } \boldsymbol{p}_{query} = \begin{pmatrix}\boldsymbol{x}_{query}\\ \boldsymbol{0}\end{pmatrix}, \boldsymbol{E}_i = \begin{pmatrix}\boldsymbol{x}_i & \boldsymbol{0} & \cdots & \boldsymbol{0}\\ \boldsymbol{y}_{i,K} & \boldsymbol{0} & \cdots & \boldsymbol{0}\end{pmatrix}\end{aligned} \quad (8)$$

$\boldsymbol{P} \in \mathbb{R}^{(2d_\mathcal{X})\times(l_{ts}K+1)}$, $\boldsymbol{E}_i \in \mathbb{R}^{(2d_\mathcal{X})\times K}$ for $i \in [l_{ts}]$. Note that in the ICL setting, $\boldsymbol{E}_i$ only has input $\boldsymbol{x}_i$ and the K-step output $\boldsymbol{y}_{i,K}$ but does not include any intermediate labels. We pad zeros in $\boldsymbol{E}_i$ so that its dimension is the same as $\boldsymbol{E}_i$ in (1) for the inference with the same model as for CoT. The ICL output is $\boldsymbol{v} = \arg\min_{\boldsymbol{u}\in\mathcal{Y}} \frac{1}{2}\|F(\Psi; \boldsymbol{P}) - \boldsymbol{u}\|^2$, following (5). The **ICL generalization error** is

$$\bar{R}_{ICL,\boldsymbol{x}_{query}\sim\mathcal{D}',f\in\mathcal{T}'}^f(\Psi) = \mathbb{E}_{\boldsymbol{x}_{query}\sim\mathcal{D}'}\left[\mathbb{1}[\boldsymbol{z}_K \neq \boldsymbol{v}]\right], \quad (9)$$

which measures the error between the one-step reasoning output and the final step label.

3. Main Theoretical Insights

We consider the setup that the model is trained using samples generated from tasks in $\mathcal{T}$ that operate on M orthonormal training-relevant (TRR) patterns, while both CoT and ICL are evaluated on tasks in $\mathcal{T}'$ that operate on M' orthonormal testing-relevant (TSR) patterns that belong to the span of TRR patterns. We consider the general setup that the context examples in the prompt for CoT and ICL testing are both noisy, i.e., TSR patterns with additive noise, and partially inaccurate, i.e., the reasoning in some examples contains incorrect steps. Our main insights are as follows.

P1. Training Dynamics of Nonlinear Transformer towards CoT. We theoretically analyze the training dynamics on a one-layer single-head attention-only Transformer to acquire the CoT generalization ability and characterize the required number of training samples and iterations. Theorem 1 shows that to learn a model with guaranteed CoT ability, the required number of context examples in each training sample, the total number of training samples, and the number of training iterations are all linear in α^{-1}, where α is the fraction of context examples with inputs that share the same TRR patterns as the query. This is consistent with the intuition that the CoT performance is enhanced if more context examples are similar to the query. Moreover, the attention values of the learned model are proved to be concentrated on testing context examples that share similar input TSR patterns as the testing query during each of the reasoning steps (Proposition 1), which is an important property that leads to the success of the CoT generalization.

P2. Guaranteed CoT Generalization. To achieve zero CoT error on tasks in $\mathcal{T}'$ with the learned model, Theorem 2 shows that the required number of context examples, where noise and errors are present, for task f in the testing prompt is proportional to $(\alpha'\tau^f\rho^f)^{-2}$, where α' is the fraction of context examples with inputs that share the same TSR patterns as the query, the constant τ^f in $(0,1)$ measures the fraction of accurate context examples, and a larger constant ρ^f in $(0,1)$ reflects a higher reasoning accuracy in each step of the examples. This result formally characterizes the intuition that more accurate context examples and more similar examples to the query improve the CoT accuracy.

P3. CoT outperforms ICL. In Theorem 3, We theoretically show that the required number of testing context examples for ICL to be successful has a similar form to that for CoT in Theorem 2, but with an additional requirement (Condition 1) that the fraction of correct input-label examples in the testing prompt must be dominant. Because not all testing cases satisfy this requirement, our result unveils the reason why CoT provides one explanation for why CoT sometimes outperforms ICL.

The formal characterizations of theoretical results are in Section C in the Appendix. We also discuss the proof sketch in Appendix D. We further introduce our numerical experiments in Appendix E.

4. Transformers Implement CoT by Attending to the Most Similar Examples in Every Step

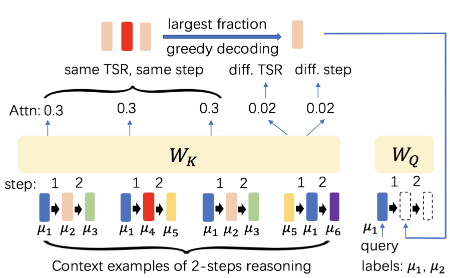


Figure 1. Concentration of attention weights for CoT inference.

In this section, we characterize the key mechanism of a properly trained one-layer Transformer to implement CoT on a K-steps reasoning task via training dynamics analysis of the attention layer, as demonstrated in Figure 1. This is different from the mechanism study in (Li et al., 2023d; Feng et al., 2023) by constructing a model that can conduct CoT. We have the following proposition for the trained model.

Proposition 1. Let $\mathcal{S}_k^*$ denote the index set of the context columns of the testing prompt $\boldsymbol{P}$ in (4) that (a) correspond to the k-th step in a context example and (b) share the same TSR pattern in the k-th input as the k-th input $\boldsymbol{v}_{k-1}$ of the query, $k \in [K]$. Given a trained model that satisfies conditions (i) to (iii) of Theorem 1 and (22) and (23) after T iterations, we have

$$\begin{aligned} &\sum_{i\in\mathcal{S}_k^*} \text{softmax}(\tilde{\boldsymbol{p}}_i^\top \boldsymbol{W}^{(T)} \tilde{\boldsymbol{q}}_k) \geq 1-\epsilon, \\ &\text{where } \tilde{\boldsymbol{p}}_i = \boldsymbol{p}_i + \boldsymbol{c}_{(i \mod K)}, \tilde{\boldsymbol{q}}_k = \boldsymbol{q}_k + \boldsymbol{c}_k, \end{aligned} \tag{10}$$

with $\boldsymbol{q}_k$ defined in (6). Moreover, for any $f \in \mathcal{T}'$, the k-th step output $\boldsymbol{v}_k$ given $\boldsymbol{x}_{query} = \boldsymbol{\mu}_i'$ satisfies,

$$\boldsymbol{v}_k = f_k \circ \cdots \circ f_1(\boldsymbol{\mu}_i'). \tag{11}$$

Proposition 1 first illustrates that, when conducting the k-th step reasoning of the query for any $k \in [K']$, the trained model assigns dominant attention weights on the prompt columns that are also the k-th step reasoning of examples and share the same TSR pattern in the k-th step input as the query. Then, given a sufficient number of testing context examples by (23), it is ensured that the fraction of the correct TSR pattern is the largest in the output of each step by (15). Subsequently, the generation by greedy decoding (5) is correct in each step, leading to a successful CoT generalization.

5. Conclusion, Limitations, and Future Works

This paper theoretically analyzes the training dynamics of Transformers with nonlinear attention, together with the CoT generalization ability of the resulting model on new tasks with noisy and partially inaccurate context examples. We quantitatively characterize and compare the required conditions for the success of CoT and ICL. Although based on a simplified Transformer model and reasoning tasks operating on patterns, this work deepens the theoretical understanding of the CoT mechanism. Future directions include designing efficient prompt-generating methods for CoT and analyzing LLM reasoning on a more complicated data model.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Ahn, K., Cheng, X., Daneshmand, H., and Sra, S. Transformers learn to implement preconditioned gradient descent for in-context learning. *arXiv preprint arXiv:2306.00297*, 2023.

Akyürek, E., Schuurmans, D., Andreas, J., Ma, T., and Zhou, D. What learning algorithm is in-context learning? investigations with linear models. In *The Eleventh International Conference on Learning Representations*, 2023.

Bai, Y., Chen, F., Wang, H., Xiong, C., and Mei, S. Transformers as statisticians: Provable in-context learning with in-context algorithm selection. *arXiv preprint arXiv:2306.04637*, 2023.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33: 1877–1901, 2020.

Chen, S. and Li, Y. Provably learning a multi-head attention layer. *arXiv preprint arXiv:2402.04084*, 2024.

Chen, S., Sheen, H., Wang, T., and Yang, Z. Training dynamics of multi-head softmax attention for in-context learning: Emergence, convergence, and optimality. *arXiv preprint arXiv:2402.19442*, 2024.

Cheng, X., Chen, Y., and Sra, S. Transformers implement functional gradient descent to learn non-linear functions in context. *arXiv preprint arXiv:2312.06528*, 2023.

Cui, Y., Ren, J., He, P., Tang, J., and Xing, Y. Superiority of multi-head attention in in-context linear regression. *arXiv preprint arXiv:2401.17426*, 2024.

Deora, P., Ghaderi, R., Taheri, H., and Thrampoulidis, C. On the optimization and generalization of multi-head attention. *arXiv preprint arXiv:2310.12680*, 2023.

Ding, N., Levinboim, T., Wu, J., Goodman, S., and Soricut, R. CausalLM is not optimal for in-context learning. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=guRNebwZBb.

Feng, G., Zhang, B., Gu, Y., Ye, H., He, D., and Wang, L. Towards revealing the mystery behind chain of thought: a theoretical perspective. *Advances in Neural Information Processing Systems*, 36, 2023.

Garg, S., Tsipras, D., Liang, P. S., and Valiant, G. What can transformers learn in-context? a case study of simple function classes. *Advances in Neural Information Processing Systems*, 35:30583–30598, 2022.

Guo, T., Hu, W., Mei, S., Wang, H., Xiong, C., Savarese, S., and Bai, Y. How do transformers learn in-context beyond simple functions? a case study on learning with representations. *arXiv preprint arXiv:2310.10616*, 2023.

Huang, Y., Cheng, Y., and Liang, Y. In-context convergence of transformers. In *NeurIPS 2023 Workshop on Mathematics of Modern Machine Learning*, 2023.

Huang, Y., Wen, Z., Chi, Y., and Liang, Y. Transformers provably learn feature-position correlations in masked image modeling. *arXiv preprint arXiv:2403.02233*, 2024.

Ildiz, M. E., Huang, Y., Li, Y., Rawat, A. S., and Oymak, S. From self-attention to markov models: Unveiling the dynamics of generative transformers. *arXiv preprint arXiv:2402.13512*, 2024.

Jelassi, S., Sander, M., and Li, Y. Vision transformers provably learn spatial structure. *Advances in Neural Information Processing Systems*, 35:37822–37836, 2022.

Li, H., Wang, M., Liu, S., and Chen, P.-Y. A theoretical understanding of shallow vision transformers: Learning, generalization, and sample complexity. In *The Eleventh International Conference on Learning Representations*, 2023a. URL https://openreview.net/forum?id=jClGv3Qjhb.

Li, H., Wang, M., Ma, T., Liu, S., ZHANG, Z., and Chen, P.-Y. What improves the generalization of graph transformer? a theoretical dive into self-attention and positional encoding. In *NeurIPS 2023 Workshop: New Frontiers in Graph Learning*, 2023b. URL https://openreview.net/forum?id=BaxFC3z9R6.

Li, H., Wang, M., Lu, S., Cui, X., and Chen, P.-Y. Training nonlinear transformers for efficient in-context learning: A theoretical learning and generalization analysis. *arXiv preprint arXiv:2402.15607*, 2024a.

Li, H., Wang, M., Zhang, S., Liu, S., and Chen, P.-Y. Learning on transformers is provable low-rank and sparse: A one-layer analysis. *arXiv preprint arXiv:2406.17167*, 2024b.

Li, Y., Li, Y., and Risteski, A. How do transformers learn topic structure: Towards a mechanistic understanding. *arXiv preprint arXiv:2303.04245*, 2023c.

Li, Y., Sreenivasan, K., Giannou, A., Papailiopoulos, D., and Oymak, S. Dissecting chain-of-thought: Compositionality through in-context filtering and learning. *Advances in Neural Information Processing Systems*, 36, 2023d.

Li, Y., Huang, Y., Ildiz, M. E., Rawat, A. S., and Oymak, S. Mechanics of next token prediction with self-attention. In *International Conference on Artificial Intelligence and Statistics*, pp. 685–693. PMLR, 2024c.

Li, Z., Liu, H., Zhou, D., and Ma, T. Chain of thought empowers transformers to solve inherently serial problems. In *The Twelfth International Conference on Learning Representations*, 2024d. URL https://openreview.net/forum?id=3EWTEy9MTM.

Liu, Y., Zhang, K., Li, Y., Yan, Z., Gao, C., Chen, R., Yuan, Z., Huang, Y., Sun, H., Gao, J., et al. Sora: A review on background, technology, limitations, and opportunities of large vision models. *arXiv preprint arXiv:2402.17177*, 2024.

Makkuva, A. V., Bondaschi, M., Girish, A., Nagle, A., Jaggi, M., Kim, H., and Gastpar, M. Attention with markov: A framework for principled analysis of transformers via markov chains. *arXiv preprint arXiv:2402.04161*, 2024.

Merrill, W. and Sabharwal, A. The expressive power of transformers with chain of thought. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=NjNGlPh8Wh.

Mohri, M., Rostamizadeh, A., and Talwalkar, A. *Foundations of machine learning*. MIT press, 2018.

Nichani, E., Damian, A., and Lee, J. D. How transformers learn causal structure with gradient descent. *arXiv preprint arXiv:2402.14735*, 2024.

OpenAI. Gpt-4 technical report. *OpenAI*, 2023.

Oymak, S., Rawat, A. S., Soltanolkotabi, M., and Thrampoulidis, C. On the role of attention in prompt-tuning. *arXiv preprint arXiv:2306.03435*, 2023.

Park, J., Park, J., Xiong, Z., Lee, N., Cho, J., Oymak, S., Lee, K., and Papailiopoulos, D. Can mamba learn how to learn? a comparative study on in-context learning tasks. *arXiv preprint arXiv:2402.04248*, 2024.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, pp. 8748–8763. PMLR, 2021.

Tarzanagh, D. A., Li, Y., Thrampoulidis, C., and Oymak, S. Transformers as support vector machines. *arXiv preprint arXiv:2308.16898*, 2023a.

Tarzanagh, D. A., Li, Y., Zhang, X., and Oymak, S. Max-margin token selection in attention mechanism. *CoRR*, 2023b.

Tian, Y., Wang, Y., Chen, B., and Du, S. Scan and snap: Understanding training dynamics and token composition in 1-layer transformer. *arXiv preprint arXiv:2305.16380*, 2023a.

Tian, Y., Wang, Y., Zhang, Z., Chen, B., and Du, S. S. Joma: Demystifying multilayer transformers via joint dynamics of mlp and attention. In *Conference on Parsimony and Learning (Recent Spotlight Track)*, 2023b.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023a.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023b.

Vershynin, R. Introduction to the non-asymptotic analysis of random matrices. *arXiv preprint arXiv:1011.3027*, 2010.

Von Oswald, J., Niklasson, E., Randazzo, E., Sacramento, J., Mordvintsev, A., Zhmoginov, A., and Vladymyrov, M. Transformers learn in-context by gradient descent. In *International Conference on Machine Learning*, pp. 35151–35174. PMLR, 2023.

Wang, L., Xu, W., Lan, Y., Hu, Z., Lan, Y., Lee, R. K.-W., and Lim, E.-P. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2609–2634, 2023.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Xia, F., Chi, E., Le, Q. V., Zhou, D., et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35: 24824–24837, 2022.

Wen, K., Dang, X., and Lyu, K. Rnns are not transformers (yet): The key bottleneck on in-context retrieval. *arXiv preprint arXiv:2402.18510*, 2024.

Wu, J., Zou, D., Chen, Z., Braverman, V., Gu, Q., and Bartlett, P. L. How many pretraining tasks are needed for in-context learning of linear regression? *arXiv preprint arXiv:2310.08391*, 2023.

Yang, K., Ackermann, J., He, Z., Feng, G., Zhang, B., Feng, Y., Ye, Q., He, D., and Wang, L. Do efficient transformers really save computation? *arXiv preprint arXiv:2402.13934*, 2024.

Zhang, R., Frei, S., and Bartlett, P. L. Trained transformers learn linear models in-context. *arXiv preprint arXiv:2306.09927*, 2023a.

Zhang, Z., Zhang, A., Li, M., and Smola, A. Automatic chain of thought prompting in large language models. In *The Eleventh International Conference on Learning Representations*, 2023b.

Zhou, D., Schärli, N., Hou, L., Wei, J., Scales, N., Wang, X., Schuurmans, D., Cui, C., Bousquet, O., Le, Q. V., et al. Least-to-most prompting enables complex reasoning in large language models. In *The Eleventh International Conference on Learning Representations*, 2023.

APPENDIX

A. Related Works

Expressive power of CoT (Li et al., 2023d) proves the existence of a Transformer that can learn a multi-layer perceptron (MLP). They interpret CoT as first filtering important tokens and then making predictions by ICL. They also establish the required number of context examples for a desired prediction with the constructed Transformer. (Feng et al., 2023; Li et al., 2024d; Merrill & Sabharwal, 2024) show that Transformers with CoT are more expressive than Transformers without CoT. (Yang et al., 2024; Wen et al., 2024) show the superiority of standard Transformers in some reasoning tasks compared with recurrent neural networks and linear Transformers.

Theoretical analysis of ICL As a simplified one-step version of CoT, ICL has gained much attention from the theoretical community. (Garg et al., 2022; Akyürek et al., 2023; Bai et al., 2023; Guo et al., 2023) demonstrate that Transformers are expressive to conduct many machine learning algorithms in context. (Akyürek et al., 2023; Von Oswald et al., 2023; Ahn et al., 2023; Cheng et al., 2023; Ding et al., 2024) especially show the existence of Transformers to implement gradient descent and its variants with different input prompts. (Zhang et al., 2023a; Huang et al., 2023; Wu et al., 2023; Li et al., 2024a) explore the training dynamics and generalization of ICL on single-attention Transformers. (Cui et al., 2024; Chen et al., 2024) provably show the superiority of multi-head attention over single-head attention to achieve ICL ability.

Training and Generalization of Transformers There have been several recent works about the optimization and generalization analysis of Transformers. (Jelassi et al., 2022; Li et al., 2023c; Oymak et al., 2023; Li et al., 2023a;b; Huang et al., 2024; Li et al., 2024b) study the generalization of one-layer Transformers by assuming spatial association, semantic/contextual structure, or the majority voting of tokens in the data. (Oymak et al., 2023; Tarzanagh et al., 2023b;a; Tian et al., 2023a;b; Li et al., 2024c; Ildiz et al., 2024; Nichani et al., 2024; Makkuva et al., 2024) investigate the training dynamics or loss landscape of Transformers for the next token prediction by assuming infinitely long input sequences, causal structure/Markov Chain of data, or a proper prediction head. (Deora et al., 2023; Chen & Li, 2024) analyze the optimization and generalization of multi-head attention networks.

B. Training Algorithms

B.1. Training Algorithm

For simplicity of analysis, we let $\boldsymbol{W}=\boldsymbol{W}_K^\top\boldsymbol{W}_Q$ and $\boldsymbol{W}_V=(\boldsymbol{0}_{d_\mathcal{X}\times d_\mathcal{X}}\ \boldsymbol{I}_{d_\mathcal{X}}\ \boldsymbol{0}_{d_\mathcal{X}\times d_\mathcal{E}})$ as (Jelassi et al., 2022; Huang et al., 2023; Zhang et al., 2023a; Huang et al., 2024). Let $\{\boldsymbol{c}_k\}_{k=1}^K$ be a set of orthonormal vectors. The model is trained using stochastic gradient descent (SGD) with step size η with batch size B, summarized in Algorithm 1 in Appendix B. Each entry of $\boldsymbol{W}^{(0)}$ is generated from $\mathcal{N}(0,\xi^2)$ for a tiny $\xi>0$. Model parameters $\boldsymbol{W}_V$ and $\boldsymbol{a}$ are fixed during the training. The fraction of prompts with $\boldsymbol{z}_{k-1}$ as the query input is $1/K$ for any $k\in[K]$ in each batch.

The training algorithm is summarized as in Algorithm 1.

Algorithm 1 Training with Stochastic Gradient Descent (SGD)

1: **Hyperparameters:** The step size η, the number of iterations T, batch size B.

2: **Initialization:** Let $\boldsymbol{W}=\boldsymbol{W}_K^\top\boldsymbol{W}_Q$ and $\boldsymbol{W}_V=(\boldsymbol{0}_{d_\mathcal{X}\times d_\mathcal{X}}\ \boldsymbol{I}_{d_\mathcal{X}}\ \boldsymbol{0}_{d_\mathcal{X}\times d_\mathcal{E}})$. Each entry of $\boldsymbol{W}^{(0)}$ is generated from $\mathcal{N}(0,\xi^2)$ for a small constant $\xi>0$. $\boldsymbol{W}_V$ and $\boldsymbol{a}$ are fixed during the training.

3: **Training by SGD:** For each iteration, we independently sample $\boldsymbol{x}_{query}\sim\mathcal{D}$, $f\in\mathcal{T}_{tr}$ to form a batch of training prompt and labels $\{\boldsymbol{P}^n, z^n\}_{n\in\mathcal{B}_t}$ as introduced in Section C.1. Each TRR pattern is sampled equally likely in each batch. For each $t=0,1,\cdots,T-1$

$$\boldsymbol{W}^{(t+1)}=\boldsymbol{W}^{(t)}-\eta\cdot\frac{1}{B}\sum_{n\in\mathcal{B}_t}\nabla_{\boldsymbol{W}^{(t)}}\ell(\Psi^{(t)};\boldsymbol{P}^n,\boldsymbol{z}^n). \tag{12}$$

4: **Output:** $\boldsymbol{W}^{(T)}$.

We then summarize the algorithm of the CoT inference introduced in Section 2.2 as in Algorithm 2.

Algorithm 2 Inference with Chain-of-Thought (CoT)

Input: $\boldsymbol{z}_0 = \boldsymbol{v}_0 = \boldsymbol{x}_{query}$, $\boldsymbol{P}_0$, and $\boldsymbol{P}_1$.
for $k = 1, \cdots, K-1$, **do**

$$\text{Compute } \boldsymbol{v}_k \text{ by greedy decoding in (5). Then update } \boldsymbol{P}_k \text{ and } \boldsymbol{P}_{k+1} \text{ by (6).} \tag{13}$$

end for
Output: $\boldsymbol{v}_1, \boldsymbol{v}_2, \cdots, \boldsymbol{v}_{K-1}$, and $\boldsymbol{v}_K$ by (5).

C. Theoretical Results

We first introduce the formulation of data and tasks in Section C.1. Sections C.2, C.3, and C.4, respectively characterize the training analysis of the Transformer and generalization using CoT and ICL with the trained model.

C.1. The Formulation of Data and Tasks

Training data and tasks: Consider M training-relevant (TRR) patterns $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \cdots, \boldsymbol{\mu}_M$, which form an orthonormal set $\mathcal{M} = \{\boldsymbol{\mu}_i\}_{i=1}^M$. $M = \Theta(d), M \leq d$. $\boldsymbol{\mu}_i \perp \boldsymbol{c}_k$ for $i \in [M'], k \in [K]$.

Every training prompt $\boldsymbol{P}$ in (1) contains the query and training examples from the same training task f in the set of training tasks $\mathcal{T}$. Specifically, each training task f is a composition of K functions $f = f_K \circ \cdots \circ f_2 \circ f_1$ where each function f_k belongs to a function set $\mathcal{F}$. The k-th step label of the query is $\boldsymbol{z}_k = f_k(\boldsymbol{z}_{k-1})$ given the k-th step input $\boldsymbol{z}_{k-1}$ with $\boldsymbol{z}_k \in \mathcal{M}$, $k \in [K]$. Moreover, the k-th step label of the i-th ($i \in [l_{tr}]$) context example is $\boldsymbol{y}_{i,k} = f_k(\boldsymbol{y}_{i,k-1})$ given the $k-1$th step input $\boldsymbol{y}_{i,k-1}, k \in [K]$ with $\boldsymbol{x}_i, \boldsymbol{y}_{i,k} \in \mathcal{M}$, where $\boldsymbol{y}_{i,0} := \boldsymbol{x}_i$. We assume that $f_k(\boldsymbol{x}) \neq f_{k'}(\boldsymbol{x}')$ if and only if either $\boldsymbol{x} \neq \boldsymbol{x}'$ or $f_k \neq f_{k'}$.

Training prompt: Consider a training prompt $\boldsymbol{P}$ on task $f \in \mathcal{T}$ defined in (1) with the query input $\boldsymbol{z}_{k-1}, k \in [K]$. Let $\alpha \in (0, 1-c]$ for some constant $c > 0$[1] denote the fraction of context examples with input sharing the same TRR pattern as the query input.

Testing task and query: Consider M' testing-relevant (TSR) patterns $\boldsymbol{\mu}'_1, \boldsymbol{\mu}'_2, \cdots, \boldsymbol{\mu}'_M$, which form an orthonormal set $\mathcal{M}' = \{\boldsymbol{\mu}'_i\}_{i=1}^{M'}$. $M' \leq M$. We also have $\boldsymbol{\mu}'_i \perp \boldsymbol{c}_k$ for $i \in [M'], k \in [K]$. Let $\mathcal{T}'$ denote the set of testing tasks, which all operate on patterns in $\mathcal{M}'$ rather than $\mathcal{M}$ in training tasks in $\mathcal{T}$. Every testing task $f = f_K \circ \cdots f_2 \circ f_1 \in \mathcal{T}'$ is a composition of K functions. The reasoning for the testing query is considered to be *noiseless* and *accurate*. That means,

$$\boldsymbol{z}_k \in \mathcal{M}' \text{ for all } k \in \{0\} \cup [K], \text{ and } \boldsymbol{z}_k = f_k(\boldsymbol{z}_{k-1}), \boldsymbol{z}_0 = \boldsymbol{x}_{query}.$$

Testing prompt: We consider the general setup that testing examples are *noisy* and *erroneous*. By noisy examples, we mean all inputs and outputs of each step are noisy versions of TSR patterns, i.e.,

$$\boldsymbol{x}_i, \boldsymbol{y}_{i,k} \in \{\boldsymbol{b} \in \mathbb{R}^d | \boldsymbol{b} = \boldsymbol{\mu}'_j + \boldsymbol{\delta}, j \in [M'], \boldsymbol{\delta} \perp \mathcal{M}', \|\boldsymbol{\delta}\| \leq \sqrt{2}/2\}, \tag{14}$$

with noise $\boldsymbol{\delta} \neq 0$ for $i \in [Kl_{ts}^f]$, $k \in [K]$. Denote TSR : $\mathbb{R}^d \mapsto \mathbb{Z}^+$ as a function that outputs the index of the TSR pattern of the noisy input. We consider the case that at least an α' fraction of context examples where the TSR pattern of the input $\boldsymbol{y}_{s,1}, s \in [l_{ts}^f]$ is the same as $\boldsymbol{x}_{query}$.

By erroneous examples, we mean that the reasoning steps in test examples may contain errors. To formally model this, we define the **step-wise transition matrices** $\{\boldsymbol{A}_k^f\}_{k=1}^K \in \mathbb{R}^{M' \times M'}$ such that $\boldsymbol{A}_k^f$ represents the reasoning probabilities of step k in test examples. Specifically, there exists some constant ρ^f in $(0, 1)$ such that for all $s \in [l_{ts}^f]$, $k \in [K]$, the i, j-th entry of $\boldsymbol{A}_k^f$ satisfies

$$\begin{aligned} &A_{k(i,j)}^f = \Pr(\text{TSR}(\boldsymbol{y}_{s,k}) = j | \text{TSR}(\boldsymbol{y}_{s,k-1}) = i), \\ \text{and } &A_{k(i,j^*)}^f \geq 1/(1-\rho^f) \cdot A_{k(i,j)}^f, \forall j \in [M'], \text{ where } \boldsymbol{\mu}'_{j^*} = f_k(\boldsymbol{\mu}'_i), \end{aligned} \tag{15}$$

Note that (15) indicates that for any given k, in the k-th reasoning step of the test example, the k-th step output is a noisy version of the true label with the highest probability, which guarantees that the examples are overall informative in the

[1]This is to prevent the trivial case that the model only learns the positional encoding but not the TRR patterns when α becomes arbitrarily close to 1.

k-th step. This requirement is intuitive because otherwise, these examples would overall provide inaccurate information on the k-th step reasoning. Moreover, (15) models the general case that, with some probability, the k-step reasoning is inaccurate in the examples. ρ^f is referred to as the **primacy** of the step-wise transition matrices. ρ^f reflects the difference in the probability of correct reasoning and incorrect reasoning in each step, and a larger ρ^f indicates a larger probability of accurate reasoning.

Let $\boldsymbol{B}^f = \prod_{k=1}^K \boldsymbol{A}_k^f$ be the K**-step transition matrix**. Then $\boldsymbol{B}_{(i,j)}^f$ is the probability that the K-th step output is a noisy version of $\boldsymbol{\mu}_j'$, when the input is a noisy version of $\boldsymbol{\mu}_i'$ in the testing example. We similarly define ρ_o^f in $(0,1)$ as the primacy of $\boldsymbol{B}^f$, where

$$B_{(i,j^*)}^f \geq 1/(1-\rho_o^f)\cdot B_{(i,j)}^f,\ \forall j\in[M'],\ j^*=\arg\max_{j\in[M']} B_{(i,j)}^f. \tag{16}$$

C.2. The Sample Complexity Analysis of the Training Stage

We first characterize the convergence and the testing performance of the model during the training stage with sample complexity analysis in Theorem 1.

Theorem 1. For any $\epsilon > 0$, when (i) the number of context examples in every training sample is

$$l_{tr} \geq \Omega(\alpha^{-1}), \tag{17}$$

(ii) the number of iterations satisfies

$$T \geq \Omega(\eta^{-1}\alpha^{-1}K^3\log\frac{K}{\epsilon} + \eta^{-1}MK(\alpha^{-1}+\epsilon^{-1})), \tag{18}$$

and (iii) the training tasks and samples are selected such that every TRR pattern is equally likely in each training batch[2] with batch size $B \geq \Omega(\max\{\epsilon^{-2}, M\}\cdot\log M)$, the step size $\eta < 1$ and $N = BT$ samples, then with a high probability, the returned model guarantees

$$\mathbb{E}_{\boldsymbol{x}_{query}\in\mathcal{M}, f\in\mathcal{T}}\left[\ell(\Psi;\boldsymbol{P},\boldsymbol{z})\right] \leq \mathcal{O}(\epsilon). \tag{19}$$

Theorem 1 indicates that with long enough training prompts and a sufficient number of iterations and samples for training, a one-layer Transformer can achieve a diminishing loss of $\mathcal{O}(\epsilon)$ on data following the same distribution as training examples. The results indicate that (i) the required number of context examples is proportional to α^{-1}; (ii) the required number of iterations and samples increases as M and α^{-1} increases. As a sanity check, these bounds are consistent with the intuition that it will make the training stage more time- and sample-consuming if the number of TRR patterns increases or the fraction of prompt examples that share the same TRR pattern as the query decreases.

C.3. CoT generalization guarantee

In this section, we first define two quantities, τ^f, and τ_o^f for each testing task $f\in\mathcal{T}'$ based on the formulation of testing data and tasks in Section C.1. These two quantities are used to characterize the CoT and ICL generalization in Theorems 2 and 3, respectively.

Definition C.1. For $f = f_K\circ\cdots f_1\in\mathcal{T}'$, we define the **min-max trajectory transition probability** as:

$$\tau^f = \min_{i\in[M']}\prod_{k=1}^K A^f_{k(\text{TSR}(f_{k-1}\circ\cdots f_0(\boldsymbol{\mu}_i')),\text{TSR}(f_k\circ\cdots f_0(\boldsymbol{\mu}_i')))},\ \text{where } f_0(\boldsymbol{\mu}_i') := \boldsymbol{\mu}_i', \forall\, i\in[M'], \tag{20}$$

which measures the minimum probability of the most probable K-step reasoning trajectory over the initial TSR pattern. We also define the **min-max input-label transition probability** as

$$\tau_o^f = \min_{i\in[M']}\max_{j\in[M']} B_{i,j}^f, \tag{21}$$

which measures the minimum probability of the most probable output over the initial TSR pattern.

Theorem 2 (CoT generalization). Given a trained model that satisfies conditions (i) to (iii) in Theorem 1, as long as (iv)

$$\boldsymbol{\mu}_j' \in \text{span}(\boldsymbol{\mu}_1,\boldsymbol{\mu}_2,\cdots,\boldsymbol{\mu}_M), \tag{22}$$

for $j\in[M']$, and (v) the number of testing examples for every task $f\in\mathcal{T}'$ is

$$l_{ts}^f \geq \Omega((\alpha'\tau^f\rho^f)^{-2}\log M), \tag{23}$$

we have $\bar{R}^f_{CoT,\boldsymbol{x}_{query}\in\mathcal{M}',f\in\mathcal{T}'}(\Psi) = 0$.

[2]This condition is to ensure a balanced gradient update among all TRR patterns, as used in (Li et al., 2024a) for ICL.

Remark 1. Theorem 2 characterizes the sufficient conditions for a trained one-layer Transformer to generate all K-steps reasoning correctly by CoT for a task f in $\mathcal{T}'$. First, the TSR patterns of a new task in $\mathcal{T}'$ should be linear combinations of TRR patterns in the training tasks in $\mathcal{T}$. Second, the number of context examples should be in the order of ${\alpha'}^{-2}$, ${\rho_s^f}^{-2}$, and ${\tau^f}^{-2}$. One can equivalently interpret the decrease in the number of required context examples to achieve zero CoT error as an improvement of the CoT accuracy with fixed context length. Then, when the fraction α' of contexts where the TSR pattern of the first step input is the same as the query increases, the contexts become more informative for the query. Thus, the CoT accuracy increases. When ρ^f and τ^f increase, the reasoning labels in the context examples are more likely to be accurate based on their definitions in (15) and (20), then the CoT accuracy is improved.

C.4. ICL Generalization and Comparison with CoT

Because only input-label pairs are used as context examples without intermediate reasoning steps for ICL, then the input-label pairs in context examples should be accurate on average. Otherwise, the context examples are not informative about the task and will lead to the failure of ICL. We formulate this requirement as Condition 1.

Condition 1. For the testing task $f = f_K \circ \cdots \circ f_1 \in \mathcal{T}'$, we have that for any $i \in [M']$,

$$\text{TSR}(f(\boldsymbol{\mu}'_i)) = \arg\max_{j\in[M']} B^f_{(i,j)}. \tag{24}$$

Condition 1 requires that in a context example, if the input TSR is $\boldsymbol{\mu}'_i$, then the output TSR needs to be $f(\boldsymbol{\mu}'_i)$ with the largest probability over all other TSR patterns. It is intuitive that the success of ICL requires this condition. Note that although (15) indicates that, $A^f_k(i, j^*)$ achieves the largest value for all j when $\boldsymbol{\mu}'_{j^*} = f_k(\boldsymbol{\mu}'_i)$ for every k and i, (15) does not always lead to (24). One example that Condition 1 may not hold is shown in Figure 4 in Section E .

Our result of the ICL generalization is stated as follows.

Theorem 3 (ICL generalization). Given a trained model that satisfies conditions (i) to (iii) of Theorem 1 and (22), for the testing task $f \in \mathcal{T}'$,

a. if Condition 1 does not hold, then $\bar{R}^f_{ICL,\boldsymbol{x}_{query}\in\mathcal{M}',f\in\mathcal{T}'}(\Psi) \geq \Omega(1)$;

b. if Condition 1 holds, we have $\bar{R}^f_{ICL,\boldsymbol{x}_{query}\in\mathcal{M}',f\in\mathcal{T}'}(\Psi) = 0$, as long as the number of testing examples is

$$l^f_{ts} \geq \Omega((\alpha'\tau^f_o\rho^f_o)^{-2}\log M). \tag{25}$$

Remark 2 (Comparison between CoT and ICL). Theorem 3(a) formally states that, Condition 1 is necessary for a successful ICL generalization. Because Condition 1 is not required for CoT generalization, CoT performs better than ICL if Condition 1 fails[3]. Theorem 3(b) characterizes that when Condition 1 holds, a desired ICL generalization needs a testing prompt length linear in ${\alpha'}^{-2}$, ${\rho^f_o}^{-2}$, and ${\tau^f_o}^{-2}$ for the testing task $f \in \mathcal{T}'$. This result is the counterpart of the requirement (23) for the CoT generalization, indicating that more context examples with the same TSR pattern as the query and more accurate context examples improve ICL generalization.

Ref. (Li et al., 2023d) also shows the advantage of CoT over ICL to learn MLP functions, but in a different setting from ours, where our studied tasks operate on patterns. More importantly, this paper characterizes the CoT and ICL performance theoretically when the testing task has a distribution shift from training tasks (TRR patterns to TSR patterns), and the testing examples contain errors, while (Li et al., 2023d) only empirically evaluates the CoT and ICL performance with noisy examples.

D. An Overview of the Proof

The technical challenges of the proof are concentrated on Theorem 1, where the property of the trained model is derived. The proof of Theorem 1 is built upon three Lemmas, which characterize the **two stages of the training dynamics**. Specifically, Lemmas F.5 and F.6 show that if a training prompt $\boldsymbol{P}$ includes the first k steps of the reasoning query, then the attention weights on columns of $\boldsymbol{P}$ with a different step from the query decrease to be close to zero in the first stage. Lemma F.7

[3]Our insight of the comparison between CoT and ICL still holds when we evaluate CoT generalization only by the final step output. This is because a successful CoT generalization in Theorem 2 on all reasoning steps already ensures a satisfactory CoT generalization on the final step.

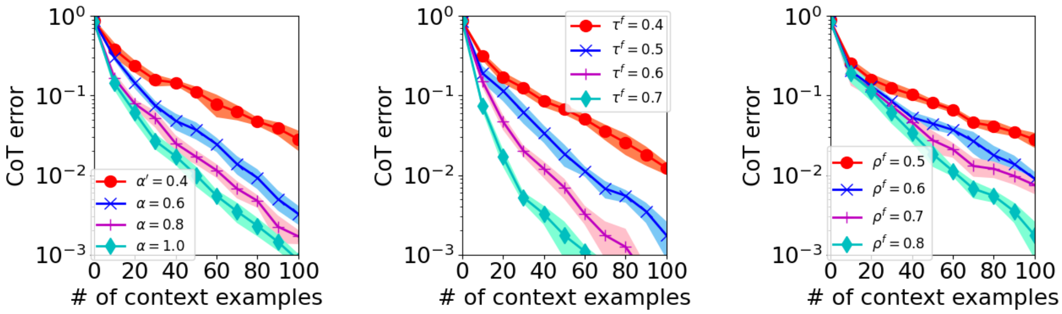
computes the gradient updates in the second stage, where the attention weights on columns in $\boldsymbol{P}$ that correspond to step k and have the same TRR pattern as the query gradually become dominant. Theorem 1 unveils this training process by showing the required number of training iterations and sample complexity.

To prove Theorem 2, we first compute the required number of context examples for the new task $f \in \mathcal{T}'$ so that by concentration inequalities, the number of context examples with accurate TSR is larger than examples with inaccurate TSR patterns in all K reasoning steps with high probability. Then, due to the linear correlation between TSR and TRR patterns (22), we also show that the trained Transformer can attend to context columns with the same TSR pattern as the query. Therefore, the model can make the correct generation in each step. Theorem 3 follows a similar proof idea to Theorem 2, with the difference that the trained model predicts output directly from the input query following $\boldsymbol{B}^f$ instead of using K reasoning steps following $\boldsymbol{A}_k^f, k \in [K]$ in CoT. Therefore, Condition 1 is required for the success of ICL generalization.

E. Numerical Experiments

Data Generation and Model setup. We use synthetic data generated following Sections 2 and C.1. Let $d_{\mathcal{X}} = 30$, $M = 20$, $M' = 10$, $\alpha = 0.4$. We consider 3-steps tasks for training and testing, i.e., $K = 3$. A reasoning task f is generated by first sampling a set of numbers of permutations $\{p_i\}_{i=1}^M$ with $p_i \in [M]$ and then let $f_k(\boldsymbol{\mu}_{p_i}) = \boldsymbol{\mu}_{p_{((i+k) \mod M)}}$ for $i \in [M], k, j \in [K]$. The testing noise level is set to be 0.2 for any examples and $f \in \mathcal{T}'$. The learning model is a one-layer single-head Transformer defined in (2). We set $\tau^f = 0.5$, $\rho^f = 0.8$, $\alpha' = 0.8$ for CoT testing if not otherwise specified. All the experiments are conducted on a single NVIDIA RTX A5000 GPU.

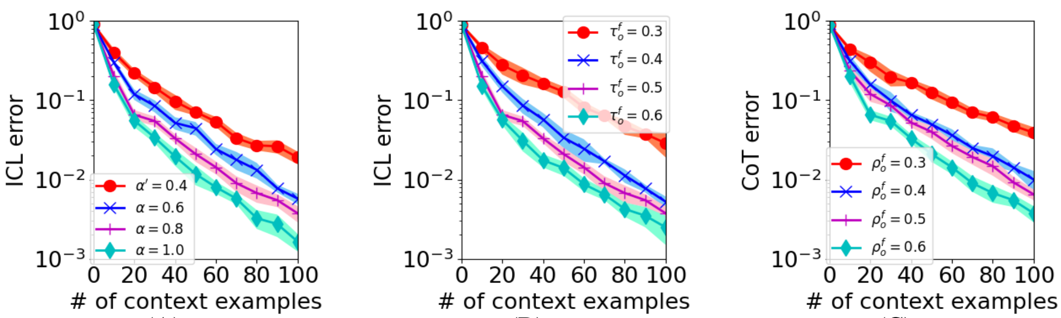
Experiments on the generalization of CoT. We first verify the required number of context examples for a desired CoT generalization. We investigate the impact of α', τ^f, and ρ^f by varying one and fixing the other two. Figure 2 illustrates that more testing examples are needed when α', τ^f, or ρ^f is small, which verifies the trend of the lower bound of l_{ts}^f in (23).



(A) (B) (C)

Figure 2. CoT testing error with different (A) α' (B) τ^f (C) ρ^f.

Experiments on the generalization of ICL and a comparison with CoT. We then verify the ICL generalization with the trained model. We vary τ_o^f and ρ_o^f by changing τ^f and ρ^f. Figure 2 indicates that more testing examples are required when α', τ_o^f, or ρ_o^f is small, which is consistent with our bound in (25). We then consider the case where $\tau_o^f = 0.4$ and $\rho_o^f = 0.1$ so that the generated testing prompt may not satisfy Condition 1 depending on the specific choices of A_k^f's. Figure 4 shows that when Condition 1 holds, the ICL testing error decreases if the number of contexts increases. However, when Condition 1 fails, the ICL testing error remains large, irrespective of the number of contexts.



(A) (B) (C)

Figure 3. ICL testing error with different (A) α' (B) τ_o^f (C) ρ_o^f.

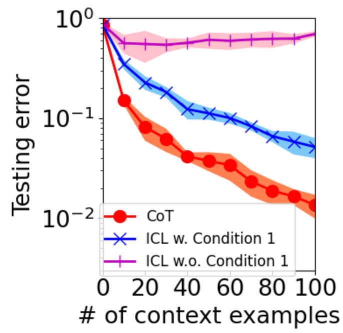


Figure 4. Comparison between CoT and ICL w./w.o. Condition 1

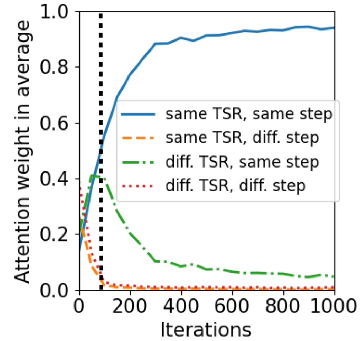


Figure 5. Training dynamics of Transformers for CoT

Experiments on the training dynamics of CoT. In Figure 5, we compute the total attention weights on four types of testing context columns along the training, which are contexts with the same (or different) TSR pattern and in the same (or different) step as the query. The result shows that the attention weights on contexts that share the same TSR pattern and in the same step as the query increase along the training and converge to around 1. This verifies the mechanism formulated in (10). Meanwhile, Figure 5 also justifies the two-stage training dynamics proposed in Section D, where we add a black vertical dashed line to demonstrate the stage transition boundary. We observe that the attention weights on context columns with a different step, i.e., the red and yellow curves, decrease to zero in the first stage. Then, the attention weights on contexts with the same TSR pattern and the same step as the query, i.e., the blue curve, increase to 1 in the second stage.

F. Preliminaries

We first summarize the notations we use in this paper in Table 1.

Lemma F.1 (Multiplicative Chernoff bounds, Theorem D.4 of (Mohri et al., 2018))**.** *Let $X_1, \cdots, \boldsymbol{X}_m$ be independent random variables drawn according to some distribution $\mathcal{D}$ with mean p and support included in $[0,1]$. Then, for any $\gamma \in [0, \frac{1}{p} - 1]$, the following inequality holds for $\hat{p} = \frac{1}{m}\sum_{i=1}^m X_i$:*

$$\Pr(\hat{p} \geq (1+\gamma)p) \leq e^{-\frac{mp\gamma^2}{3}}, \tag{26}$$

$$\Pr(\hat{p} \leq (1-\gamma)p) \leq e^{-\frac{mp\gamma^2}{2}}. \tag{27}$$

Definition F.2 ((Vershynin, 2010))**.** We say X is a sub-Gaussian random variable with sub-Gaussian norm $K > 0$, if $(\mathbb{E}|X|^p)^{\frac{1}{p}} \leq K\sqrt{p}$ for all $p \geq 1$. In addition, the sub-Gaussian norm of X, denoted $\|X\|_{\psi_2}$, is defined as $\|X\|_{\psi_2} = \sup_{p\geq 1} p^{-\frac{1}{2}}(\mathbb{E}|X|^p)^{\frac{1}{p}}$.

Lemma F.3 ((Vershynin, 2010) Proposition 5.1, Hoeffding's inequality)**.** *Let $X_1, X_2, \cdots, X_N$ be independent centered sub-gaussian random variables, and let $K = \max_i \|\boldsymbol{X}_i\|_{\psi_2}$. Then for every $\boldsymbol{a} = (a_1, \cdots, a_N) \in \mathbb{R}^N$ and every $t \geq 0$, we have*

$$\Pr\Big(\Big|\sum_{i=1}^N a_i X_i\Big| \geq t\Big) \leq e \cdot \exp\Big(-\frac{ct^2}{K^2\|\boldsymbol{a}\|^2}\Big), \tag{28}$$

where $c > 0$ is an absolute constant.

Definition F.4. Define that for $\tilde{\boldsymbol{p}}_i$ that shares the same TRR/TSR pattern and in the same step as the query,

$$p_n(t) = \sum_i \text{softmax}({\tilde{\boldsymbol{p}}_i^n}^\top \boldsymbol{W}^{(t)} \tilde{\boldsymbol{p}}_{query}^n). \tag{29}$$

Lemma F.5. *Given the SGD training scheme described in Section B.1, $B \geq \Omega(M \log M)$, and $l_{tr} \geq \Omega(\alpha^{-1})$, we have the*

Table 1. Summary of Notations

Notations	Annotation
$\boldsymbol{x}_i, \boldsymbol{y}_{i,k}, \boldsymbol{x}_{query}, \boldsymbol{z}_k$	$\boldsymbol{x}_i$ is the input to the first step of a reasoning example. $\boldsymbol{y}_{i,k}$ is the k-th step output label of $\boldsymbol{x}_i$. $\boldsymbol{x}_{query}$ is the query input. $\boldsymbol{z}_k$ the k-th step output label of $\boldsymbol{x}_{query}$. $k \in [K]$.
$\boldsymbol{P}, \boldsymbol{p}_{query}, \boldsymbol{E}_i, \boldsymbol{Q}_k, \boldsymbol{v}_k$	$\boldsymbol{P}$ is a training or testing prompt that consists of multiple training or testing examples and a query. The last column of $\boldsymbol{P}$ is denoted by $\boldsymbol{p}_{query}^n$, which is the query of $\boldsymbol{P}$. $\boldsymbol{E}_i$ is the i-th context example of $\boldsymbol{P}$. $\boldsymbol{Q}_k$ is the first k steps of the reasoning query. $k \in [K]$. $\boldsymbol{v}_k$ is the k-th step generation by CoT. $k \in [K]$.
$\boldsymbol{c}_i, \tilde{\boldsymbol{p}}_i, \tilde{\boldsymbol{p}}_{query}$	$\boldsymbol{c}_i$ is the positional encoding for the i-th column of the input sequence. $\tilde{\boldsymbol{p}}_i = \boldsymbol{p}_i + \boldsymbol{c}_i$, where $\boldsymbol{p}_i$ is the i-th column of $\boldsymbol{P}$. $\tilde{\boldsymbol{p}}_{query}$ is the $\boldsymbol{p}_i$ of the query column.
$F(\Psi; \boldsymbol{P}), \ell(\Psi; \boldsymbol{P}^n, \boldsymbol{z}^n)$	$F(\Psi; \boldsymbol{P}^n)$ is the Transformer output for $\boldsymbol{P}$ with Ψ as the parameter. $\ell(\Psi; \boldsymbol{P}^n, \boldsymbol{z}^n)$ is the loss function value given $\boldsymbol{P}^n$ and the corresponding label $\boldsymbol{z}^n$.
$\boldsymbol{\mu}_i \in \mathcal{M}, \boldsymbol{\mu}_i' \in \mathcal{M}', \text{TSR}(\cdot)$	$\boldsymbol{\mu}_i$ is the i-th training-relevant (TRR) pattern for $i \in [M]$. $\boldsymbol{\mu}_i'$ is the i-th testing-relevant (TSR) pattern for $i \in [M']$. $\mathcal{M}$ and $\mathcal{M}'$ are the set of TRR and TSR patterns, respectively. TSR$(\cdot)$ is a function that outputs the index of the TSR pattern of the noisy input.
f_k, f	f is the task function with $f = f_K \circ \cdots f_2 \circ f_1$ for a K-steps reasoning. f_k is the k-th step task function.
$\mathcal{T}, \mathcal{T}', \mathcal{D}, \mathcal{D}'$	$\mathcal{T}$ is the distribution of training tasks, while $\mathcal{T}'$ is the distribution of testing tasks. $\mathcal{D}$ is the training data distribution. $\mathcal{D}'$ is the testing data distribution.
α, α'	α (or α') is the fraction of context examples with input sharing the same TRR (or TSR) pattern as the query.
$\boldsymbol{A}_k^f, \boldsymbol{B}_k^f$	$\boldsymbol{A}_k^f$ is the step-wise transition matrix at the k-th step for the task f, $k \in [K]$. $\boldsymbol{B}_k^f$ is the K-steps transition matrix of the task f.
$\tau^f, \tau_o^f, \rho^f, \rho_o^f$	τ^f is the min-max trajectory transition probability for task f. τ_o^f is the min-max input-label transition probability for task f. ρ^f and ρ_o^f are primacy of the step-wise transition matrices and the K-steps transition matrix, respectively.
$\mathcal{S}_k^*, \mathcal{B}_b$	The index set of context columns of the prompt that correspond to the k-th step of the example and share the same TSR pattern in the $(k-1)$-th output as the $(k-1)$-th output $\boldsymbol{v}_{k-1}$ of the query. $\mathcal{B}_b$ is the SGD batch at the b-th iteration.
l_{tr}	l_{tr} is the universal number of training context examples.
l_{ts}^f	l_{ts} is the number of testing context examples of the task f.
$\mathcal{O}(), \Omega(), \Theta(), \gtrsim, \lesssim$	We follow the convention that $f(x) = O(g(x))$ (or $\Omega(g(x))$, $\Theta(g(x))$)) means that $f(x)$ increases at most, at least, or in the order of $g(x)$, respectively. $f(x) \gtrsim g(x)$ (or $f(x) \lesssim g(x)$) means that $f(x) \geq \Omega(g(x))$ (or $f(x) \lesssim \mathcal{O}(g(x))$).

following results. When $\mathcal{O}(\eta^{-1}\alpha^{-2}K^3 \log \frac{K}{\epsilon}) \geq t \geq 1$, for any $\boldsymbol{p}$ as a column of context examples in (1), we have

$$\begin{aligned}
&\tilde{\boldsymbol{p}}^\top \eta \frac{1}{B} \sum_{n \in \mathcal{B}_b} \frac{\partial \ell(\Psi; \boldsymbol{P}^n, \boldsymbol{z}^n)}{\partial \boldsymbol{W}^{(t)}} \tilde{\boldsymbol{p}} \\
\leq& \frac{\eta}{B} \sum_{n \in \mathcal{B}_b} (\frac{1}{KM}(1-p_n(t))^2(-4p_n(t)(1+\frac{\alpha^2}{K^2}) + \frac{\alpha^2}{K^2}(1+\frac{2(K-1)}{K})) - \frac{\alpha^2}{K^3}(1-p_n(t))^2).
\end{aligned} \tag{30}$$

For any $\tilde{\mathbf{p}}'$ that shares the same TRR pattern and a different positional encoding as $\tilde{\mathbf{p}}$, we have

$$\begin{aligned}
&\frac{\eta}{B}\sum_{n\in\mathcal{B}_b}(\frac{1}{KM}(-4-(3K-2)(1-p_n(t))(1+\frac{\alpha^2}{K^2}))p_n(t)(1-p_n(t))+\frac{\alpha^2}{K^3}(1-p_n(t))^2)\\
\leq&\tilde{\boldsymbol{p}'}^\top\eta\frac{1}{B}\sum_{n\in\mathcal{B}_b}\frac{\partial\ell(\Psi;\boldsymbol{P}^n,\boldsymbol{z}^n)}{\partial\boldsymbol{W}^{(t)}}\tilde{\boldsymbol{p}}\\
\leq&\frac{\eta}{B}\sum_{n\in\mathcal{B}_b}(\frac{1}{KM}(-4-(3K-2)(1-p_n(t))(1+\frac{\alpha^2}{K^2}))p_n(t)(1-p_n(t))+\frac{1}{K}p_n(t)(1-p_n(t))^2\\
&\cdot(1+\frac{\alpha^2}{K^2})).
\end{aligned}\tag{31}$$

For any $\tilde{\mathbf{p}'}$ that shares a different TRR pattern but the same positional encoding as $\tilde{\mathbf{p}}$, we have

$$\begin{aligned}
&\eta\cdot\frac{1}{B}\sum_{n\in\mathcal{B}_b}(\frac{1}{KM}(-\frac{\alpha^2}{K^2}+(K-1+\frac{(2K-1)\alpha^2}{K^2})p_n(t))(1-p_n(t))^2-(1-p_n(t))^2\frac{\alpha^2}{K^3}\\
&+\frac{1}{K}\cdot(1-p_n(t))^2(-p_n(t)+(1-p_n(t))\frac{\alpha^2}{K^2}))\\
\leq&\tilde{\boldsymbol{p}'}^\top\eta\frac{1}{B}\sum_{n\in\mathcal{B}_b}\frac{\partial\ell(\Psi;\boldsymbol{P}^n,\boldsymbol{z}^n)}{\partial\boldsymbol{W}^{(t)}}\tilde{\boldsymbol{p}}\\
\leq&\eta\cdot\frac{1}{B}\sum_{n\in\mathcal{B}_b}(\frac{1}{KM}(-\frac{\alpha^2}{K^2}+(K-1+\frac{(2K-1)\alpha^2}{K^2})p_n(t))(1-p_n(t))^2-(1-p_n(t))^2\frac{\alpha^2}{K^3}).
\end{aligned}\tag{32}$$

For any $\tilde{\mathbf{p}'}$ that shares a different TRR pattern and a different positional encoding from $\tilde{\mathbf{p}}$, we have

$$\begin{aligned}
&\eta\cdot\frac{1}{B}\sum_{n\in\mathcal{B}_b}(\frac{1}{KM}p_n(t)(1-p_n(t))^2(1+\frac{(2-K)\alpha^2}{K^2})+(1-p_n(t))^2\cdot\frac{\alpha^2}{K^3})\\
\leq&\tilde{\boldsymbol{p}'}^\top\eta\frac{1}{B}\sum_{n\in\mathcal{B}_b}\frac{\partial\ell(\Psi;\boldsymbol{P}^n,\boldsymbol{z}^n)}{\partial\boldsymbol{W}^{(t)}}\tilde{\boldsymbol{p}}\\
\leq&\eta\cdot\frac{1}{B}\sum_{n\in\mathcal{B}_b}(\frac{1}{KM}p_n(t)(1-p_n(t))^2(2-K+\frac{(2-K)\alpha^2}{K^2})+(1-p_n(t))^2p_n(t)(1+\frac{\alpha^2}{K^2})\cdot\frac{1}{K}).
\end{aligned}\tag{33}$$

Lemma F.6. *Given the SGD training scheme described in Section B.1, $B\geq\Omega(M\log M)$, and $l_{tr}\geq\Omega(\alpha^{-1})$, and*

$$t\gtrsim T_1:=\eta^{-1}\alpha^{-2}K^3\log\frac{K}{\epsilon},\tag{34}$$

we have that if $\boldsymbol{p}_{query}$ is in the k-th step,

$$\sum_{i\in\mathcal{S}_{[K]\setminus k}}\textit{softmax}(\tilde{\boldsymbol{p}}_i^\top\boldsymbol{W}^{(t)}\tilde{\boldsymbol{p}}_{query})\leq\epsilon\tag{35}$$

where $\mathcal{S}_{[K]\setminus k}$ means the index set of context columns that are not in the k-th step.

Lemma F.7. *Given the SGD training scheme described in Section B.1, $B\geq\Omega(M\log M)$, and $l_{tr}\geq\Omega(\alpha^{-1})$, we have the following results. When $t\geq T_1=\eta^{-1}\alpha^{-2}K^3\log\frac{K}{\epsilon}$, for any $\boldsymbol{p}$ as a column of context examples in (1), we have*

$$\tilde{\boldsymbol{p}}^\top\eta\frac{1}{B}\sum_{n\in\mathcal{B}_b}\frac{\partial\ell(\Psi;\boldsymbol{P}^n,\boldsymbol{z}^n)}{\partial\boldsymbol{W}^{(t)}}\tilde{\boldsymbol{p}}\leq-\frac{\eta}{2MB}\sum_{n\in\mathcal{B}_b}4p_n(t)(1-p_n(t))^2.\tag{36}$$

For any $\tilde{\mathbf{p}'}$ that shares the same TRR pattern and a different positional encoding as $\tilde{\mathbf{p}}$, we have

$$\left|\tilde{\boldsymbol{p}'}^\top\eta\frac{1}{B}\sum_{n\in\mathcal{B}_b}\frac{\partial\ell(\Psi;\boldsymbol{P}^n,\boldsymbol{z}^n)}{\partial\boldsymbol{W}^{(t)}}\tilde{\boldsymbol{p}}\right|\leq\eta\epsilon.\tag{37}$$

For any $\tilde{\boldsymbol{p}'}$ that shares a different TRR pattern but the same positional encoding as $\tilde{\boldsymbol{p}}$, we have

$$\left|\tilde{\boldsymbol{p}'}^\top \eta \frac{1}{B}\sum_{n\in\mathcal{B}_b}\frac{\partial \ell(\Psi;\boldsymbol{P}^n,\boldsymbol{z}^n)}{\partial \boldsymbol{W}^{(t)}}\tilde{\boldsymbol{p}}\right| \leq \frac{\eta}{2BM}\sum_{n\in\mathcal{B}_b} p_n(b)(1-p_n(b))^2. \tag{38}$$

For any $\tilde{\boldsymbol{p}'}$ that shares a different TRR pattern and a different positional encoding from $\tilde{\boldsymbol{p}}$, we have

$$\left|\tilde{\boldsymbol{p}'}^\top \eta \frac{1}{B}\sum_{n\in\mathcal{B}_b}\frac{\partial \ell(\Psi;\boldsymbol{P}^n,\boldsymbol{z}^n)}{\partial \boldsymbol{W}^{(t)}}\tilde{\boldsymbol{p}}\right| \leq \eta\epsilon. \tag{39}$$

G. Proof of Main Theorems

G.1. Proof of Theorem 1

Proof. By the condition in Lemma F.5, we have that

$$B \geq \Omega(M\log M). \tag{40}$$

We know that there exists gradient noise caused by imbalanced TRR patterns in each batch. Then, by Hoeffding's inequality (28),

$$\begin{aligned}&\Pr\left(\left\|\frac{1}{|\mathcal{B}_b|}\sum_{n\in\mathcal{B}_b}\frac{\partial \ell(\Psi;\boldsymbol{P}^n,z^n)}{\partial \boldsymbol{W}} - \mathbb{E}\left[\frac{\partial \ell(\Psi;\boldsymbol{P}^n,z^n)}{\partial \boldsymbol{W}}\right]\right\| \geq \left|\mathbb{E}\left[\frac{\partial \ell(\Psi;\boldsymbol{P}^n,z^n)}{\partial \boldsymbol{W}}\right]\epsilon\right)\\ \leq& e^{-B\epsilon^2}\leq M^{-C},\end{aligned} \tag{41}$$

if $B \gtrsim \epsilon^{-2}\log M$. Therefore, we require

$$B \gtrsim \max\{\epsilon^{-2}, M\}\log M. \tag{42}$$

By Lemma F.7 and Definition F.4, for $\tilde{\boldsymbol{p}_i^n}$ that share the same TRR pattern and the same positional encoding of $\tilde{\boldsymbol{p}}_{query}^n$,

$$\frac{p_n(t+1)}{|\mathcal{S}_1^n|} = \text{softmax}(\tilde{\boldsymbol{p}_i^n}^\top \boldsymbol{W}^{(t+1)}\tilde{\boldsymbol{p}}_{query}^n) \geq \frac{1}{l}\cdot\frac{1}{\frac{\alpha}{K}+(1-\frac{1}{K})\cdot\epsilon+(\frac{1}{K}-\frac{\alpha}{K})e^{-u}}, \tag{43}$$

where by (159),

$$u \gtrsim \frac{\eta}{KM}\sum_{b=0}^{t}(1-p_n(b))^2 p_n(b). \tag{44}$$

For $\tilde{\boldsymbol{p}_i^n}$ that only share the same positional encoding of $\tilde{\boldsymbol{p}}_{query}^n$,

$$\text{softmax}(\tilde{\boldsymbol{p}_i^n}^\top \boldsymbol{W}^{(t+1)}\tilde{\boldsymbol{p}}_{query}^n) \geq \frac{1}{l}\cdot\frac{1}{\frac{\alpha}{K}e^u+(1-\frac{1}{K})\cdot\epsilon+(\frac{1}{K}-\frac{\alpha}{K})}. \tag{45}$$

Therefore, to make the attention weights between $\tilde{\boldsymbol{p}}_{query}^n$ and $\tilde{\boldsymbol{p}_i^n}$ that share the same TRR pattern and the same positional encoding dominant, we need a large enough u. When $1-p_n(b)\geq\Omega(1)$, we have

$$t \leq T_2 := \eta^{-1}KM\alpha^{-1}. \tag{46}$$

When $1-p_n(b)\leq O(1)$,

$$p_n(t+1) = \frac{e^u}{e^u+\frac{1-\alpha}{\alpha}} \gtrsim 1-\frac{1-\alpha}{\alpha}e^{-u}, \tag{47}$$

and

$$1-p_n(t+1) \geq \frac{1-\alpha}{\alpha e^u+(1-\alpha)} \gtrsim \frac{1-\alpha}{\alpha}e^{-u}. \tag{48}$$

Then, we prove that when t is large enough, $u(t) \geq \frac{1}{2}\log\frac{\eta(1-\alpha)^2 t}{\alpha^2 M}$. We show it by induction. Suppose that the conclusion holds when $t=t_0$, then

$$\begin{aligned} u(t) \geq & \frac{\eta}{KM}\sum_{b=0}^{t_0}(1-p_n(b))^2 p_n(b) + \frac{\eta}{KM}(1-p_n(t))^2 p_n(t) \\ \geq & \frac{1}{2}\log\frac{(1-\alpha)^2 t}{2\alpha^2 KM} + \frac{\eta}{KM}(1-p_n(t))^2 p_n(t) \\ \geq & \frac{1}{2}\log\frac{\eta(1-\alpha)^2(t+1)}{\alpha^2 KM}, \end{aligned} \tag{49}$$

where the last step is by

$$\frac{1}{2}\log(1+\frac{1}{t}) \leq \frac{1}{2t} \leq \frac{\eta}{KM}\cdot(\frac{1-\alpha}{\alpha})^2 e^{-\log\frac{\eta(1-\alpha)^2 t}{\alpha^2 KM}}. \tag{50}$$

To make $(1-p_n(t))^2 < \epsilon$, we need

$$(\frac{1-\alpha}{\alpha})^2 e^{-2u} \leq \epsilon. \tag{51}$$

Then, we get

$$u \geq \frac{1}{2}\log\frac{1}{\epsilon} + \log\frac{1-\alpha}{\alpha}. \tag{52}$$

Therefore, by

$$\frac{1}{2}\log\frac{\eta t}{KM} + \log\frac{1-\alpha}{\alpha} \geq \frac{1}{2}\log\frac{1}{\epsilon} + \log\frac{1-\alpha}{\alpha}, \tag{53}$$

we finally obtain

$$t \geq T_3 := \eta^{-1}\epsilon^{-1}KM. \tag{54}$$

For $\tilde{\boldsymbol{p}_i^n}$ that shares the same TSR pattern as the query, we have that when $t=T_1$,

$$\tilde{\boldsymbol{p}_i^n}^\top \boldsymbol{W}^{(t)}\tilde{\boldsymbol{p}}_{query}^n \geq \log\frac{K}{\epsilon}. \tag{55}$$

When $t = T_1+T_2+T_3$,

$$\tilde{\boldsymbol{p}_i^n}^\top \boldsymbol{W}^{(t)}\tilde{\boldsymbol{p}}_{query}^n \geq \Theta(1)\cdot\log\frac{K}{\epsilon} = \Theta(\log\frac{K}{\epsilon}). \tag{56}$$

Then,

$$\begin{aligned} T :=& T_1+T_2+T_3 \\ =& \Theta(\eta^{-1}\alpha^{-1}K^3\log\frac{K}{\epsilon} + \eta^{-1}MK(\alpha^{-1}+\epsilon^{-1})). \end{aligned} \tag{57}$$

Therefore,

$$\mathbb{E}_{\boldsymbol{x}_{query}\sim\mathcal{D}, f\in\mathcal{T}}\left[\ell(\Psi;\boldsymbol{P},\boldsymbol{z})\right] \leq \mathcal{O}(\epsilon). \tag{58}$$

□

G.2. Proof of Theorem 2

Proof. We know that α' is the fraction of examples that share the same TSR pattern as the query. We need that in each step, the number of examples that share the same TSR pattern as the current step of the query is at least 1. Note that the probability of examples where each reasoning step produces the most probable output is

$$\prod_{k=1}^K A^f_{k(\text{TSR}(f_{k-1}\circ\cdots f_0(\boldsymbol{\mu}_i')),\text{TSR}(f_k\circ\cdots f_0(\boldsymbol{\mu}_i')))}, \text{ where } f_0(\boldsymbol{\mu}_i') := \boldsymbol{\mu}_i', \forall\, i\in[M'], \tag{59}$$

where the input to the first step has the TSR pattern $\boldsymbol{\mu}_i'$. Define $m_{k(i)}$ as the TSR pattern in the k-th step output of the i-th context example by the transition matrix defined in 15. Consider that the TSR pattern of the k-th step label of the testing query is $\boldsymbol{\mu}'_{q_k}$, which is also the most probable k-th step output of the k-th step of a certain $\boldsymbol{x}_i$ with TSR$(\boldsymbol{x}_i) = $ TSR$(\boldsymbol{x}_{query}) = q_0$. Let the TSR pattern of another reasoning process, where for a certain first-step input $\boldsymbol{x}_i$ with TSR$(\boldsymbol{x}) = $ TSR$(\boldsymbol{x}_{query}) = q_0$, the k-th step output is the most probable for $k\in[K']\backslash\{h\}$, while the h-th step output is the second probable. Denote the

TSR pattern of the k-th step output of $\boldsymbol{x}_i$ following this process as $\boldsymbol{\mu}'_{u_k}$ with $u_0 = q_0$. By the Chernoff bound of Bernoulli distribution in Lemma F.1, we can obtain

$$\begin{aligned}&\Pr\left(\frac{1}{l_{ts}}\sum_{i=1}^{l_{ts}}\mathbb{1}[m_{k(i)}=\boldsymbol{\mu}'_{q_k},\forall k\in[K']]\leq (1-\rho_s^f/2)\alpha'\prod_{k=1}^{K'}A^f_{k(q_{k-1},q_k)}\right)\\ \leq & e^{-l_{ts}(\rho_s^f)^2\alpha'\prod_{k=1}^{K'}A^f_{k(q_{k-1},q_k}}=M^{-C},\end{aligned} \tag{60}$$

and by Lemma F.3,

$$\begin{aligned}&\Pr\left(\frac{1}{l_{ts}}\sum_{i=1}^{l_{ts}}\mathbb{1}[m_{k(i)}=\boldsymbol{\mu}'_{u_k},\forall k\in[K']]\geq (1-\rho_s^f/2)\alpha'\prod_{k=1}^{K'}A^f_{k(q_{k-1},q_k)}\right)\\ \leq &\Pr\left(\frac{1}{l_{ts}}\sum_{i=1}^{l_{ts}}\mathbb{1}[m_{k(i)}=\boldsymbol{\mu}'_{u_k},\forall k\in[K']]\geq \alpha'\prod_{k=1}^{K'}A^f_{k(u_{k-1},u_k)}+t_0\right)\\ \leq & e^{-l_{ts}t_0^2}=M^{-C},\end{aligned} \tag{61}$$

for some $c\in(0,1)$ and $C>0$, where the first step is by the definition of ρ_s^f in (15), and

$$t_0\lesssim \rho_s^f\alpha'\prod_{k=1}^{K'}A^f_{k(q_{k-1},q_k)}. \tag{62}$$

Hence, with a high probability,

$$\begin{aligned}l_{ts}\gtrsim&\max\{({\rho_s^f}^2\alpha'\prod_{k=1}^{K'}A^f_{k(q_{k-1},q_k)})^{-1}\log M,(\rho_s^f\alpha'\prod_{k=1}^{K'}A^f_{k(q_{k-1},q_k)})^{-2}\log M\}\\ \gtrsim&(\rho_s^f\alpha'\prod_{k=1}^{K'}A^f_{k(q_{k-1},q_k)})^{-2}\log M,\end{aligned} \tag{63}$$

such that the number of examples with the same TSR pattern as the query in each of the total K steps is at least 1. To make the above condition hold for any TSR pattern of the intermediate step of the query, we need

$$\begin{aligned}l_{ts}\gtrsim&\max_{q_k\in[M']}(\rho_s^f\alpha'\prod_{k=1}^{K'}A^f_{k(q_{k-1},q_k)})^{-2}\log M\\ =&\max_{i\in[M']}(\rho_s^f\alpha'\prod_{k=1}^{K'}A^f_{k(\text{TSR}(f_{k-1}\circ\cdots f_0(\boldsymbol{\mu}'_i)),\text{TSR}(f_k\circ\cdots f_0(\boldsymbol{\mu}'_i)))})^{-2}\log M\\ =&(\rho_s^f\alpha'\tau_s^f)^{-2}\log M.\end{aligned} \tag{64}$$

Then, we show the CoT testing error is zero by induction. In the first step, consider $\boldsymbol{x}_i=\boldsymbol{\mu}_j+\boldsymbol{\delta}_i$ such that

$$\tilde{\boldsymbol{p}}_i=\begin{pmatrix}\boldsymbol{\mu}'_j\\ \boldsymbol{y}_i\end{pmatrix}+\begin{pmatrix}\boldsymbol{\delta}_i\\ \boldsymbol{0}\end{pmatrix}+\boldsymbol{c}_i. \tag{65}$$

Since that

$$(\boldsymbol{\delta}_i^\top,0^\top)\boldsymbol{W}^{(0)}\tilde{\boldsymbol{p}}_i\lesssim\xi, \tag{66}$$

by that each entry of $\boldsymbol{W}^{(0)}$ follows $\mathcal{N}(0,\xi^2)$, and

$$(\boldsymbol{\delta}_i^\top,0^\top)\frac{\eta}{B}\sum_{n\in\mathcal{B}_b}\sum_{b=0}^{T-1}\frac{\partial\ell(\Psi;\boldsymbol{P}^n,\boldsymbol{z}^n)}{\partial\boldsymbol{W}^{(b)}}\tilde{\boldsymbol{p}}_{query}=0, \tag{67}$$

we have that for $\tilde{\boldsymbol{p}}_i$ that shares the same TSR pattern as the query,

$$\begin{aligned}
&\tilde{\boldsymbol{p}_i}^\top \boldsymbol{W}^{(T)}\tilde{\boldsymbol{p}}_{query}\\
=&\tilde{\boldsymbol{p}_i}^\top(\boldsymbol{W}^{(0)}+\frac{\eta}{B}\sum_{n\in\mathcal{B}_b}\sum_{b=0}^{T-1}\frac{\partial \ell(\Psi;\boldsymbol{P}^n,\boldsymbol{z}^n)}{\partial \boldsymbol{W}^{(b)}})\tilde{\boldsymbol{p}}_{query}\\
=&(({\boldsymbol{\mu}'_j}^\top,\boldsymbol{y}_i^\top)+\boldsymbol{c}_i^\top))(\boldsymbol{W}^{(0)}+\frac{\eta}{B}\sum_{n\in\mathcal{B}_b}\sum_{b=0}^{T-1}\frac{\partial \ell(\Psi;\boldsymbol{P}^n,\boldsymbol{z}^n)}{\partial \boldsymbol{W}^{(b)}})\tilde{\boldsymbol{p}}_{query}.
\end{aligned}\tag{68}$$

Let $\boldsymbol{\mu}'_j=\sum_{i=1}^{M'}\lambda_{j,i}\boldsymbol{\mu}_i$. Then, we have

$$\begin{aligned}
&\tilde{\boldsymbol{p}_i}^\top \boldsymbol{W}^{(T)}\tilde{\boldsymbol{p}}_{query}\\
=&((\sum_{i=1}^{M'}\lambda_{j,i}\boldsymbol{\mu}_i^\top,\boldsymbol{y}_i^\top)+\boldsymbol{c}_i^\top)(\boldsymbol{W}^{(0)}+\frac{\eta}{B}\sum_{n\in\mathcal{B}_b}\sum_{b=0}^{T-1}\frac{\partial \ell(\Psi;\boldsymbol{P}^n,\boldsymbol{z}^n)}{\partial \boldsymbol{W}^{(b)}})((\sum_{i=1}^{M'}\lambda_{j,i}\boldsymbol{\mu}_i^\top,\boldsymbol{0}^\top)+\boldsymbol{c}_1)^\top\\
=&\sum_{i=1}^{M'}\lambda_{j,i}^2((\boldsymbol{\mu}_i^\top,\boldsymbol{y}_i^\top)+\boldsymbol{c}_i^\top)(\boldsymbol{W}^{(0)}+\frac{\eta}{B}\sum_{n\in\mathcal{B}_b}\sum_{b=0}^{T-1}\frac{\partial \ell(\Psi;\boldsymbol{P}^n,\boldsymbol{z}^n)}{\partial \boldsymbol{W}^{(b)}})((\boldsymbol{\mu}_i^\top,\boldsymbol{0}^\top)+\boldsymbol{c}_1)^\top\\
&+\sum_{i\neq i'}\lambda_{j,i}\lambda_{j,i'}(\boldsymbol{\mu}_i^\top,\boldsymbol{y}_i^\top,\boldsymbol{c}_i^\top)(\boldsymbol{W}^{(0)}+\frac{\eta}{B}\sum_{n\in\mathcal{B}_b}\sum_{b=0}^{T-1}\frac{\partial \ell(\Psi;\boldsymbol{P}^n,\boldsymbol{z}^n)}{\partial \boldsymbol{W}^{(b)}})((\boldsymbol{\mu}_{i'}^\top,\boldsymbol{0}^\top)+\boldsymbol{c}_1)^\top\\
\geq&\Theta(\log\frac{K}{\epsilon})-\epsilon\\
=&\Theta(\log\frac{K}{\epsilon}),
\end{aligned}\tag{69}$$

where the second to last step is by Theorem 1. Since the gradient updates for different TRR patterns are very close to each other, we have that if $\sum_{i=1}^{M'}\lambda_{j,i}\lambda_{k,i}=0$,

$$\begin{aligned}
&\sum_{i=1}^{M'}\lambda_{j,i}\lambda_{j,i'}((\boldsymbol{\mu}_i^\top,\boldsymbol{y}_i^\top)+\boldsymbol{c}_i^\top)(\boldsymbol{W}^{(0)}+\frac{\eta}{B}\sum_{n\in\mathcal{B}_b}\sum_{b=0}^{T-1}\frac{\partial \ell(\Psi;\boldsymbol{P}^n,\boldsymbol{z}^n)}{\partial \boldsymbol{W}^{(b)}})((\boldsymbol{\mu}_i^\top,\boldsymbol{0}^\top)+\boldsymbol{c}_1)^\top\\
\lesssim&\epsilon\log\frac{K}{\epsilon}.
\end{aligned}\tag{70}$$

Hence, for $\tilde{\boldsymbol{p}}_i$ that shares a different TSR pattern with $\tilde{\boldsymbol{p}}_i$,

$$\tilde{\boldsymbol{p}_i}^\top \boldsymbol{W}^{(T)}\tilde{\boldsymbol{p}}_{query}\lesssim\epsilon\log\frac{K}{\epsilon}.\tag{71}$$

Therefore, we can derive that

$$\sum_{i\in\mathcal{S}_1^*}\text{softmax}(\tilde{\boldsymbol{p}_i}^\top \boldsymbol{W}^{(T)}\tilde{\boldsymbol{p}}_{query})\geq 1-\epsilon,\tag{72}$$

where $\mathcal{S}_1^*$ is the set of the first step of examples that share the same TSR pattern as the query. Then, the first step leads to a correct prediction with zero testing error, since that $\max_{j\in[M']}A_{k(q_0,j)}$ is the largest to make the correct prediction for $\boldsymbol{x}_{query}$ if $\boldsymbol{x}_{query}=\boldsymbol{\mu}'_{q_0}$, i.e.,

$$\boldsymbol{v}_1=f_1(\boldsymbol{\mu}'_{q_0}).\tag{73}$$

Suppose that the k-th step generates a zero testing error. Then, for the $k+1$-th step, we know that there exists $\boldsymbol{p}_j$ that shares the same TSR pattern as $\boldsymbol{v}_k$. Then, we can also derive that

$$\tilde{\boldsymbol{p}}_j^\top \boldsymbol{W}^{(T)}((\boldsymbol{v}_k^\top,\boldsymbol{0}^\top)^\top+\boldsymbol{c}_k^\top)^\top=\Theta(\log\frac{K}{\epsilon}),\tag{74}$$

and

$$\sum_{j\in\mathcal{S}_k^*}\text{softmax}(\tilde{\boldsymbol{p}}_j^\top \boldsymbol{W}^{(T)}((\boldsymbol{v}_{k-1}^\top\ \boldsymbol{v}_k^\top)^\top+\boldsymbol{c}_k^\top)^\top)\geq 1-\epsilon.\tag{75}$$

Hence, the $k+1$-th also makes the correct prediction, i.e.,

$$\boldsymbol{v}_{k+1} = f_{k+1} \circ \cdots f_1(\boldsymbol{\mu}'_{q_0}), \tag{76}$$

where $\boldsymbol{\mu}'_{q_{k+1}}$ is the TSR pattern of the $k+1$-th step input. Therefore, we show that CoT makes the correct prediction in each step as well as in the final prediction, such that

$$\bar{R}^f_{CoT,\boldsymbol{x}\in\mathcal{M}',f\in\mathcal{T}'}(\Psi) = 0. \tag{77}$$

□

G.3. Proof of Theorem 3

Proof. We know that the positional encodings are the same for the ICL inference in all examples. Hence, similar to (72), we can derive that

$$\sum_{i\in\mathcal{S}^*_K} \text{softmax}(\tilde{\boldsymbol{p}}_i^\top \boldsymbol{W}^{(T)}\tilde{\boldsymbol{p}}_{query}) \geq 1-\epsilon, \tag{78}$$

where $\mathcal{S}^*_K$ is the set of the last step output of examples that share the same TSR pattern as the last step output of the query. For $\boldsymbol{x}_{query} = \boldsymbol{\mu}'_q$, $q \in [K']$, we know that the distribution of the corresponding label $\boldsymbol{y}$ of $\boldsymbol{x}$ with TSR$(\boldsymbol{x}) = q$ follows the q-th row the K-steps transition matrix B^f. Let $F(\Psi;\boldsymbol{P}) = \sum_{i=1}^{M'} \lambda_i^{\boldsymbol{P}} \boldsymbol{\mu}'_i$. Hence, based on the output scheme of ICL as stated in Section 2.2, we have that

$$\boldsymbol{v} = \arg\min_{\boldsymbol{y}\in\mathcal{M}'} \frac{1}{2}\|F(\Psi;\boldsymbol{P}) - \boldsymbol{y}\|^2 = \boldsymbol{\mu}_{\arg\max_{i\in[M']}\lambda_i^{\boldsymbol{P}}}. \tag{79}$$

Note that the probability of examples with the most probable final output with $\boldsymbol{\mu}'_q$ as the TSR pattern of the input is

$$B_{(q,\text{TSR}(f(\boldsymbol{\mu}'_q)))}. \tag{80}$$

To ensure that the number of examples with the same TSR pattern as the query that generates the most probable output is at least 1, we compute the following,

$$\begin{aligned}
&\Pr\left(\frac{1}{l_{ts}}\sum_{i=1}^{l_{ts}} \mathbb{1}[m_i = \boldsymbol{\mu}'_{q_1}] \leq (1-\rho_o^f/2)\alpha' B_{(q,\text{TSR}(f(\boldsymbol{\mu}'_q)))}\right)\\
\leq & e^{-l_{ts}{\rho_o^f}^2\alpha' B_{(q,\text{TSR}(f(\boldsymbol{\mu}'_q)))}} = M^{-C},
\end{aligned} \tag{81}$$

for some $c\in(0,1)$ and $C>0$ by the Chernoff bound of Bernoulli distribution in Lemma F.1. Here, m_i is defined as the TSR pattern in the final output of the i-th context example by the K-steps transition matrix defined in 16. The TSR pattern of the most probable output of the testing query is $\boldsymbol{\mu}'_{q_1}$. Similarly, let the TSR pattern of the second most probable output of the testing query be $\boldsymbol{\mu}'_{q_2}$. We also have

$$\begin{aligned}
&\Pr\left(\frac{1}{l_{ts}}\sum_{i=1}^{l_{ts}} \mathbb{1}[m_i = \boldsymbol{\mu}'_{q_2}] \geq (1-\rho_o^f/2)\alpha' B^f_{(q,q_1)}\right)\\
\leq &\Pr\left(\frac{1}{l_{ts}}\sum_{i=1}^{l_{ts}} \mathbb{1}[m_i = \boldsymbol{\mu}'_{q_2}] \geq \alpha' B_{(q,q_2)} + c\cdot\rho_o^f\alpha' B^f_{(q,q_1)}\right)\\
\leq & e^{-l_{ts}{\rho_o^f}^2 c^2\alpha' B_{(q,q_1)}} = M^{-C},
\end{aligned} \tag{82}$$

by Lemma F.3 and (16) for some constant $c>0$. Therefore, to make the number of examples with the same TSR pattern in the output as the label of the query be at least 1 for any TSR pattern of the query and the output be the most probable one, we need

$$\begin{aligned}
l_{ts}^f \gtrsim & \max\{({\rho_o^f}^2\alpha' \min_{i\in[M']} B_{(i,\text{TSR}(f(\boldsymbol{\mu}'_i)))})^{-1}\log M, (\rho_o^f\alpha' \min_{i\in[M']} B_{(i,\text{TSR}(f(\boldsymbol{\mu}'_i)))})^{-2}\log M\}\\
= & (\rho_o^f\alpha'\tau_o^f)^{-2}\log M\}.
\end{aligned} \tag{83}$$

In addition, if Condition 1 holds such that the most probable output is the actual label, we can derive

$$\bar{R}^f_{ICL,\boldsymbol{x}\in\mathcal{M}',f\in\mathcal{T}'}(\Psi)=0. \tag{84}$$

When (83) holds but Condition 1 does not, we know that ICL still always produces the most probable output by the K-steps transition matrix, but such an output is not the label since Condition 1 fails. Hence,

$$\bar{R}^f_{ICL,\boldsymbol{x}\in\mathcal{M}',f\in\mathcal{T}'}(\Psi)\geq\Omega(1). \tag{85}$$

When both Condition 1 and (83) do not hold, ICL can produce multiple possible outputs with a non-trivial probability, which is decided by the distribution of the prompt instead of the K-steps transition matrix. This can be seen from that (81) and (82) both do not hold since (83) fails. Then, ICL can produce both the most probable and the second most probable output with a constant probability. Let the TSR pattern of the r-th most probable output of the testing query be $\boldsymbol{\mu}'_r$. Recall that $F(\Psi;\boldsymbol{P})=\sum_{i=1}^{M'}\lambda_i^{\boldsymbol{P}}\boldsymbol{\mu}'_i$, we then have that for some small $\epsilon>0$,

$$\lambda^{\boldsymbol{P}}_{r(q)}=\frac{|\{i\in[l_{ts}^f]:\boldsymbol{y}_i=\boldsymbol{\mu}'_r\text{ in }\boldsymbol{P}\}|}{l_{ts}^f}\pm\epsilon. \tag{86}$$

Then, by (79), the output of the query is $\boldsymbol{\mu}_{\arg\max_{r\in[M']}\lambda_r}$. Since that (83) does not hold, there exists at least a constant probability of the prompt $\boldsymbol{P}'$ with the same query as $\boldsymbol{P}$ such that

$$\lambda^{\boldsymbol{P}'}_{r}=\frac{|\{i\in[l_{ts}^f]:\boldsymbol{y}_i=\boldsymbol{\mu}'_r\text{ in }\boldsymbol{P}'\}|}{l_{ts}^f}\pm\epsilon\neq\lambda_r^{\boldsymbol{P}}, \tag{87}$$

for some $r\in[M']$. Therefore, with a constant probability, the output for the same testing query and the same testing task f varies. This leads to

$$\bar{R}^f_{ICL,\boldsymbol{x}\in\mathcal{M}',f\in\mathcal{T}'}(\Psi)\geq\Omega(1). \tag{88}$$

□

G.4. Proof of Proposition 1

Proof. This proposition is derived from the proof of Theorem 2. (10) comes from (75), while (11) comes from (76), both by induction. □

H. Proof of Lemmas

H.1. Proof of Lemma F.5

Proof.

$$\begin{aligned}
&\eta\frac{1}{B}\sum_{n\in\mathcal{B}_b}\frac{\partial\ell(\Psi;\boldsymbol{P}^n,\boldsymbol{z}^n)}{\partial\boldsymbol{W}}\\
=&\eta\frac{1}{B}\sum_{n\in\mathcal{B}_b}\frac{\partial\ell(\Psi;\boldsymbol{P}^n,\boldsymbol{z}^n)}{\partial F(\Psi;\boldsymbol{P})}\frac{\partial F(\Psi;\boldsymbol{P})}{\partial\boldsymbol{W}}\\
=&\eta\frac{1}{B}\sum_{n\in\mathcal{B}_b}(F(\Psi;\boldsymbol{P})-\boldsymbol{z}^n)^\top\sum_{i=1}^{l}\boldsymbol{W}_V\tilde{\boldsymbol{p}}_i\text{softmax}(\tilde{\boldsymbol{p}}_i{}^\top\boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\\
&\cdot(\tilde{\boldsymbol{p}}_i-\sum_{r=1}^{l}\text{softmax}(\tilde{\boldsymbol{p}}_r{}^\top\boldsymbol{W}\tilde{\boldsymbol{p}}_i)\tilde{\boldsymbol{p}}_r)\tilde{\boldsymbol{p}}_{query}^\top.
\end{aligned} \tag{89}$$

When $t=0$, we know that each entry of $\boldsymbol{W}^{(0)}$ is generated from the Gaussian distribution $\mathcal{N}(0,\xi^2)$. Then,

$$|\tilde{\boldsymbol{p}}_i{}^\top\boldsymbol{W}^{(0)}\tilde{\boldsymbol{p}}_{query}|=|\sum_{k,j}p_{i,k}p_{query,j}W^{(0)}_{k,j}|\lesssim\xi. \tag{90}$$

Hence,

$$\text{softmax}(\tilde{\boldsymbol{p}_i}^\top \boldsymbol{W}^{(0)} \tilde{\boldsymbol{p}}_{query}) \geq \frac{e^{-\Theta(\xi)}}{l \cdot e^{\Theta(\xi)}} = \frac{1}{l} \cdot e^{-\Theta(\xi)}, \tag{91}$$

$$\text{softmax}(\tilde{\boldsymbol{p}_i}^\top \boldsymbol{W}^{(0)} \tilde{\boldsymbol{p}}_{query}) \leq \frac{e^{-\Theta(\xi)}}{l \cdot e^{\Theta(\xi)}} = \frac{1}{l} \cdot e^{-\Theta(\xi)}. \tag{92}$$

We can obtain

$$F(\Psi; \boldsymbol{P}) = \sum_{i=1}^{l} \frac{e^{-\Theta(\xi)}}{l} \boldsymbol{W}_V \boldsymbol{p}_i. \tag{93}$$

Since that PE$(\cdot)$, and TRR$(\cdot)$ denote the positional encoding, and the TSR pattern of the input, respectively, we have that for $\boldsymbol{p}$,

$$\tilde{\boldsymbol{p}}^\top \tilde{\boldsymbol{p}}_{query} = \mathbb{1}[\text{TRR}(\tilde{\boldsymbol{p}}) = \text{TRR}(\tilde{\boldsymbol{p}}_{query})] + \mathbb{1}[\text{PE}(\tilde{\boldsymbol{p}}) = \text{PE}(\tilde{\tilde{\boldsymbol{p}}}_i)]. \tag{94}$$

Given lab$(\cdot)$ is the label embedding of the context as the input, we have that for $\boldsymbol{p}$,

$$\tilde{\boldsymbol{p}}^\top \tilde{\boldsymbol{p}_i} = \mathbb{1}[\text{TRR}(\tilde{\boldsymbol{p}}) = \text{TRR}(\tilde{\boldsymbol{p}_i})] + \mathbb{1}[\text{lab}(\tilde{\boldsymbol{p}}) = \text{lab}(\tilde{\boldsymbol{p}_i})] + \mathbb{1}[\text{PE}(\tilde{\boldsymbol{p}}) = \text{PE}(\tilde{\boldsymbol{p}_i})], \tag{95}$$

$$(\boldsymbol{W}_V \tilde{\boldsymbol{p}})^\top \boldsymbol{W}_V \tilde{\boldsymbol{p}_i} = \mathbb{1}[\text{lab}(\tilde{\boldsymbol{p}}) = \text{lab}(\tilde{\boldsymbol{p}_i})]. \tag{96}$$

When $t \geq 1$, we first consider the case where $\tilde{\boldsymbol{p}}$ shares the same TRR pattern and the positional encoding as $\tilde{\boldsymbol{p}}_{query}$. If $\tilde{\boldsymbol{p}}$ and $\tilde{\boldsymbol{p}}_{query}$ share the same TRR pattern, label pattern, and the positional encoding,

$$\begin{aligned} \tilde{\boldsymbol{p}}^\top (\tilde{\boldsymbol{p}_i} - \sum_{r=1}^{l} \text{softmax}(\tilde{\boldsymbol{p}_r}^\top \boldsymbol{W} \tilde{\boldsymbol{p}}_{query}) \tilde{\boldsymbol{p}_r}) \tilde{\boldsymbol{p}}_{query}^\top \tilde{\boldsymbol{p}} \geq & 2 \cdot (3 - 3p_n(t) - (1 - p_n(t))) \\ = & 4(1 - p_n(t)), \end{aligned} \tag{97}$$

and

$$\tilde{\boldsymbol{p}}^\top (\tilde{\boldsymbol{p}_i} - \sum_{r=1}^{l} \text{softmax}(\tilde{\boldsymbol{p}_r}^\top \boldsymbol{W} \tilde{\boldsymbol{p}}_{query}) \tilde{\boldsymbol{p}_r}) \tilde{\boldsymbol{p}}_{query}^\top \tilde{\boldsymbol{p}} \leq 2 \cdot (3 - 3p_n(t)) = 6(1 - p_n(t)). \tag{98}$$

When $\tilde{\boldsymbol{p}}$ and $\tilde{\boldsymbol{p}}_{query}$ only share the same positional encoding or the same TRR pattern,

$$2 - 6p_n(t) \geq \tilde{\boldsymbol{p}}^\top (\tilde{\boldsymbol{p}_i} - \sum_{r=1}^{l} \text{softmax}(\tilde{\boldsymbol{p}_r}^\top \boldsymbol{W} \tilde{\boldsymbol{p}}_{query}) \tilde{\boldsymbol{p}_r}) \tilde{\boldsymbol{p}}_{query}^\top \tilde{\boldsymbol{p}} \geq -4p_n(t). \tag{99}$$

When $\tilde{\boldsymbol{p}}$ and $\tilde{\boldsymbol{p}}_{query}$ share both different positional encodings and TRR patterns,

$$-6p_n(t) \geq \tilde{\boldsymbol{p}}^\top (\tilde{\boldsymbol{p}_i} - \sum_{r=1}^{l} \text{softmax}(\tilde{\boldsymbol{p}_r}^\top \boldsymbol{W} \tilde{\boldsymbol{p}}_{query}) \tilde{\boldsymbol{p}_r}) \boldsymbol{p}_{\tilde{qeury}}^\top \tilde{\boldsymbol{p}} \geq -2 - 4p_n(t). \tag{100}$$

Then, we consider the case where $\tilde{\boldsymbol{p}}$ only shares the same TRR pattern or the same positional encoding as $\tilde{\boldsymbol{p}_i}$. If $\tilde{\boldsymbol{p}}$ and $\tilde{\boldsymbol{p}}_{query}$ share the same TRR pattern, label pattern, and the positional encoding,

$$\begin{aligned} 3 - p_n(t) \geq \tilde{\boldsymbol{p}}^\top (\tilde{\boldsymbol{p}_i} - \sum_{r=1}^{l} \text{softmax}(\tilde{\boldsymbol{p}_r}^\top \boldsymbol{W} \tilde{\boldsymbol{p}}_{query}) \tilde{\boldsymbol{p}_r}) \tilde{\boldsymbol{p}}_{query}^\top \tilde{\boldsymbol{p}} \geq & 1 \cdot (3 - p_n(t) - (1 - p_n(t))) \\ = & 2. \end{aligned} \tag{101}$$

When $\tilde{\boldsymbol{p}}$ and $\tilde{\boldsymbol{p}}_{query}$ only share the same positional encoding or the same TRR pattern,

$$1 - p_n(t) \geq \tilde{\boldsymbol{p}}^\top (\tilde{\boldsymbol{p}_i} - \sum_{r=1}^{l} \text{softmax}(\tilde{\boldsymbol{p}_r}^\top \boldsymbol{W} \tilde{\boldsymbol{p}}_{query}) \tilde{\boldsymbol{p}_r}) \tilde{\boldsymbol{p}}_{query}^\top \tilde{\boldsymbol{p}} \geq 0. \tag{102}$$

When $\tilde{\boldsymbol{p}}$ and $\tilde{\boldsymbol{p}}_{query}$ only share both different positional encodings and TRR patterns,

$$-p_n(t) \geq \tilde{\boldsymbol{p}}^\top(\tilde{\boldsymbol{p}}_i - \sum_{r=1}^{l} \text{softmax}({\tilde{\boldsymbol{p}_r}}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\tilde{\boldsymbol{p}_r})\tilde{\boldsymbol{p}}_{query}^\top \tilde{\boldsymbol{p}} \geq -1. \tag{103}$$

Note that $-(1-p_n(t))p_n(t) + (1-p_n(t))^2\alpha^2/K^2 < 0$ for $p_n(t) \in [\alpha/K, \alpha]$. Then, when $l \geq \Omega(\alpha^{-1})$ and $\tilde{\boldsymbol{p}}$ shares the same TRR pattern and the positional encoding as $\tilde{\boldsymbol{p}}_i$,

$$\begin{aligned}
&(\sum_{i=1}^{l} \text{softmax}({\tilde{\boldsymbol{p}}_i}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\boldsymbol{W}_V\tilde{\boldsymbol{p}}_i - \boldsymbol{z}^n)^\top \sum_{i=1}^{l} \text{softmax}({\tilde{\boldsymbol{p}}_i}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\boldsymbol{W}_V\tilde{\boldsymbol{p}}_i\\
&\cdot \tilde{\boldsymbol{p}}^\top(\tilde{\boldsymbol{p}}_i - \sum_{r=1}^{l} \text{softmax}({\tilde{\boldsymbol{p}_r}}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\tilde{\boldsymbol{p}_r})\tilde{\boldsymbol{p}}_{query}^\top \tilde{\boldsymbol{p}}\\
\leq & -4p_n(t)(1-p_n(t))^2 - 4p_n(t)(1-p_n(t))^2\cdot\frac{\alpha^2}{K^2}\\
&+\frac{1}{l}(\frac{1}{K}-\frac{\alpha}{K})(-4p_n(t)) + \frac{1}{l}(\frac{1}{K}-\frac{\alpha}{K})(1-p_n(t))(-2-4p_n(t))(K-1)\\
= & -4p_n(t)(1-p_n(t))^2(1+\frac{\alpha^2}{K^2}) + \frac{2}{lK}(1-\alpha)(-(K-1)-(K+1)p_n(t)+2p_n(t)^2(K-1)).
\end{aligned} \tag{104}$$

We next consider the case where $\tilde{\boldsymbol{p}}$ shares the same TRR pattern and the different positional encoding as $\tilde{\boldsymbol{p}}_{query}$. Note that

$$\frac{2}{Kl}\cdot(1-\alpha)\cdot K(1-p_n(t)) \lesssim |(-(1-p_n(t))p_n(t) + (1-p_n(t))^2\frac{\alpha^2}{K^2})(1-p_n(t))|, \tag{105}$$

if $l \geq \Omega(\alpha^{-1})$. Then,

$$\begin{aligned}
&(\sum_{i=1}^{l} \text{softmax}({\tilde{\boldsymbol{p}}_i}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\boldsymbol{W}_V\tilde{\boldsymbol{p}}_i - \boldsymbol{z}^n)^\top \sum_{i=1}^{l} \text{softmax}({\tilde{\boldsymbol{p}}_i}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\boldsymbol{W}_V\tilde{\boldsymbol{p}}_i\\
&\cdot \tilde{\boldsymbol{p}}^\top(\tilde{\boldsymbol{p}}_i - \sum_{r=1}^{l} \text{softmax}({\tilde{\boldsymbol{p}_r}}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\tilde{\boldsymbol{p}_r})\tilde{\boldsymbol{p}}_{query}^\top \tilde{\boldsymbol{p}}\\
\leq & -0\cdot p_n(t)(1-p_n(t)) + (1-p_n(t))^2\frac{\alpha^2}{K^2}\cdot(+2) + \frac{1}{l}(\frac{1}{K}-\frac{\alpha}{K})(-(K-1))\\
= & 2(1-p_n(t))^2\frac{\alpha^2}{K^2} - \frac{K-1}{l}(\frac{1}{K}-\frac{\alpha}{K}).
\end{aligned} \tag{106}$$

We next consider the case where $\tilde{\boldsymbol{p}}$ shares the same positional encoding and the different TRR pattern as $\tilde{\boldsymbol{p}}_{query}$. Then,

$$\begin{aligned}
&(\sum_{i=1}^{l} \text{softmax}({\tilde{\boldsymbol{p}}_i}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\boldsymbol{W}_V\tilde{\boldsymbol{p}}_i - \boldsymbol{z}^n)^\top \sum_{i=1}^{l} \text{softmax}({\tilde{\boldsymbol{p}}_i}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\boldsymbol{W}_V\tilde{\boldsymbol{p}}_i\\
&\cdot \tilde{\boldsymbol{p}}^\top(\tilde{\boldsymbol{p}}_i - \sum_{r=1}^{l} \text{softmax}({\tilde{\boldsymbol{p}_r}}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\tilde{\boldsymbol{p}_r}){\tilde{\boldsymbol{p}}_i}^\top \tilde{\boldsymbol{p}}\\
\leq & 0 - (1-p_n(t))^2\frac{\alpha^2}{K^2} + \frac{1}{l}(\frac{1}{K}-\frac{\alpha}{K})(-(K-1))\\
= & -(1-p_n(t))^2\frac{\alpha^2}{K^2} - \frac{K-1}{l}(\frac{1}{K}-\frac{\alpha}{K}).
\end{aligned} \tag{107}$$

Therefore, as long as

$$l \geq \Omega(\alpha^{-1}), \tag{108}$$

we have

$$\begin{aligned}
&\tilde{\boldsymbol{p}}^\top \eta \frac{1}{B}\sum_{n\in\mathcal{B}_b}\frac{\partial \ell(\Psi;\boldsymbol{P}^n,\boldsymbol{z}^n)}{\partial \boldsymbol{W}}\boldsymbol{p}\\
=&\eta\frac{1}{B}\sum_{n\in\mathcal{B}_b}(F(\Psi;\boldsymbol{P})-\boldsymbol{z}^n)^\top\sum_{i=1}^l \boldsymbol{W}_V\tilde{\boldsymbol{p}}_i\text{softmax}(\tilde{\boldsymbol{p}}_i{}^\top\boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\\
&\cdot\tilde{\boldsymbol{p}}^\top(\tilde{\boldsymbol{p}}_i-\sum_{r=1}^l\text{softmax}(\tilde{\boldsymbol{p}}_r{}^\top\boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\tilde{\boldsymbol{p}}_r)\tilde{\boldsymbol{p}}_i{}^\top\tilde{\boldsymbol{p}}\\
\leq&\eta\frac{1}{B}\sum_{n\in\mathcal{B}_b}(\frac{1}{KM}(1-p_n(t))^2(-4p_n(t)(1+\frac{\alpha^2}{K^2})+\frac{2(K-1)\alpha^2}{K^2})\\
&\cdot+(\frac{1}{K}-\frac{1}{M})(-(1-p_n(t))^2\frac{\alpha^2}{K^2}))\\
=&\eta\cdot\frac{1}{B}\sum_{n\in\mathcal{B}_b}(\frac{1}{KM}(1-p_n(t))^2(-4p_n(t)(1+\frac{\alpha^2}{K^2})+\frac{\alpha^2}{K}(1+\frac{2(K-1)}{K}))-\frac{\alpha^2}{K^3}(1-p_n(t))^2).
\end{aligned}\tag{109}$$

We then consider the case where $\tilde{\boldsymbol{p}}'$ shares a different positional encoding and the same TRR pattern as $\tilde{\boldsymbol{p}}$. Let $\tilde{\boldsymbol{p}}$ share the same TRR pattern and the positional encoding as $\tilde{\boldsymbol{p}}_{query}$. If $\tilde{\boldsymbol{p}}'$ and $\tilde{\boldsymbol{p}}_i$ share the same TRR pattern, label pattern, and the positional encoding,

$$\begin{aligned}
2(3-p_n(t))\geq\tilde{\boldsymbol{p}'}^\top(\tilde{\boldsymbol{p}}_i-\sum_{r=1}^l\text{softmax}(\tilde{\boldsymbol{p}}_r{}^\top\boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\tilde{\boldsymbol{p}}_r)\tilde{\boldsymbol{p}}_{query}^\top\tilde{\boldsymbol{p}}\geq&2\cdot(3-p_n(t)-(1-p_n(t)))\\
=&4.
\end{aligned}\tag{110}$$

When $\tilde{\boldsymbol{p}}'$ and $\tilde{\boldsymbol{p}}_{query}$ only share the same positional encoding or the same TRR pattern,

$$2(1-p_n(t))\geq\tilde{\boldsymbol{p}'}^\top(\tilde{\boldsymbol{p}}_i-\sum_{r=1}^l\text{softmax}(\tilde{\boldsymbol{p}}_r{}^\top\boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\tilde{\boldsymbol{p}}_r)\tilde{\boldsymbol{p}}_{query}^\top\tilde{\boldsymbol{p}}\geq 0.\tag{111}$$

When $\tilde{\boldsymbol{p}}'$ and $\tilde{\boldsymbol{p}}_i$ only share both different positional encodings and TRR patterns,

$$-2p_n(t)\geq\tilde{\boldsymbol{p}'}^\top(\tilde{\boldsymbol{p}}_i-\sum_{r=1}^l\text{softmax}(\tilde{\boldsymbol{p}}_r{}^\top\boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\tilde{\boldsymbol{p}}_r)\tilde{\boldsymbol{p}}_{query}^\top\tilde{\boldsymbol{p}}\geq -2.\tag{112}$$

Then, we consider the case where $\tilde{\boldsymbol{p}}$ only shares the same TRR pattern as $\tilde{\boldsymbol{p}}_{query}$. If $\tilde{\boldsymbol{p}}'$ and $\tilde{\boldsymbol{p}}_i$ share the same TRR pattern, label pattern, and the positional encoding,

$$\begin{aligned}
3-p_n(t))\geq&\tilde{\boldsymbol{p}'}^\top(\tilde{\boldsymbol{p}}_i-\sum_{r=1}^l\text{softmax}(\tilde{\boldsymbol{p}}_r{}^\top\boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\tilde{\boldsymbol{p}}_r)\tilde{\boldsymbol{p}}_{query}^\top\tilde{\boldsymbol{p}}\\
\geq&1\cdot(3-3p_n(t)-(1-p_n(t)))=2(1-p_n(t)).
\end{aligned}\tag{113}$$

When $\tilde{\boldsymbol{p}}'$ and $\tilde{\boldsymbol{p}}_i$ only share the same positional encoding or the same TRR pattern,

$$1-p_n(t)\geq\tilde{\boldsymbol{p}'}^\top(\tilde{\boldsymbol{p}}_i-\sum_{r=1}^l\text{softmax}(\tilde{\boldsymbol{p}}_r{}^\top\boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\tilde{\boldsymbol{p}}_r)\tilde{\boldsymbol{p}}_{query}^\top\tilde{\boldsymbol{p}}\geq -2p_n(t).\tag{114}$$

When $\tilde{\boldsymbol{p}}'$ and $\tilde{\boldsymbol{p}}_i$ only share both different positional encodings and TRR patterns,

$$-p_n(t)\geq\tilde{\boldsymbol{p}'}^\top(\tilde{\boldsymbol{p}}_i-\sum_{r=1}^l\text{softmax}(\tilde{\boldsymbol{p}}_r{}^\top\boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\tilde{\boldsymbol{p}}_r)\tilde{\boldsymbol{p}}_{query}^\top\tilde{\boldsymbol{p}}\geq -1-2p_n(t).\tag{115}$$

Next, we consider the case where $\tilde{\boldsymbol{p}}$ only shares the same positional encoding as $\tilde{\boldsymbol{p}}_{query}$. If $\tilde{\boldsymbol{p}}'$ and $\tilde{\boldsymbol{p}}_i$ share the same TRR pattern, label pattern, and the positional encoding,

$$3 \geq \tilde{\boldsymbol{p}'}^\top(\tilde{\boldsymbol{p}}_i - \sum_{r=1}^{l} \text{softmax}(\tilde{\boldsymbol{p}_r}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\tilde{\boldsymbol{p}_r})\tilde{\boldsymbol{p}}_{query}^\top \tilde{\boldsymbol{p}} \geq 1 \cdot (3 - (1 - p_n(t))) = 2 + p_n(t). \tag{116}$$

When $\tilde{\boldsymbol{p}}'$ and $\tilde{\boldsymbol{p}}_i$ only share the same positional encoding or the same TRR pattern,

$$1 \geq \tilde{\boldsymbol{p}'}^\top(\tilde{\boldsymbol{p}}_i - \sum_{r=1}^{l} \text{softmax}(\tilde{\boldsymbol{p}_r}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\tilde{\boldsymbol{p}_r})\tilde{\boldsymbol{p}}_{query}^\top \tilde{\boldsymbol{p}} \geq p_n(t). \tag{117}$$

When $\tilde{\boldsymbol{p}}'$ and $\tilde{\boldsymbol{p}}_i$ only share both different positional encodings and TRR patterns,

$$0 \geq \tilde{\boldsymbol{p}'}^\top(\tilde{\boldsymbol{p}}_i - \sum_{r=1}^{l} \text{softmax}(\tilde{\boldsymbol{p}_r}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\tilde{\boldsymbol{p}_r})\tilde{\boldsymbol{p}}_{query}^\top \tilde{\boldsymbol{p}} \geq -1 + p_n(t). \tag{118}$$

Then, when $l \geq \Omega(\alpha^{-1})$ and $\tilde{\boldsymbol{p}}$ shares the same TRR pattern and the positional encoding as $\tilde{\boldsymbol{p}}_{query}$,

$$\begin{aligned}
&(\sum_{i=1}^{l} \text{softmax}(\tilde{\boldsymbol{p}_i}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\boldsymbol{W}_V\tilde{\boldsymbol{p}}_i - \boldsymbol{z}^n)^\top \sum_{i=1}^{l} \text{softmax}(\tilde{\boldsymbol{p}_i}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\boldsymbol{W}_V\tilde{\boldsymbol{p}}_i \\
&\cdot \tilde{\boldsymbol{p}'}^\top(\tilde{\boldsymbol{p}}_i - \sum_{r=1}^{l} \text{softmax}(\tilde{\boldsymbol{p}_r}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\tilde{\boldsymbol{p}_r})\tilde{\boldsymbol{p}}_{query}^\top \tilde{\boldsymbol{p}} \\
\leq& -4p_n(t)(1-p_n(t)) + \frac{1}{l}(\frac{1}{K} - \frac{\alpha}{K})(-2K).
\end{aligned} \tag{119}$$

We next consider the case where $\tilde{\boldsymbol{p}}$ shares the same TRR pattern and the different positional encoding as $\tilde{\boldsymbol{p}}_{query}$. Then, by (105),

$$\begin{aligned}
&(\sum_{i=1}^{l} \text{softmax}(\tilde{\boldsymbol{p}_i}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\boldsymbol{W}_V\tilde{\boldsymbol{p}}_i - \boldsymbol{z}^n)^\top \sum_{i=1}^{l} \text{softmax}(\tilde{\boldsymbol{p}_i}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\boldsymbol{W}_V\tilde{\boldsymbol{p}}_i \\
&\cdot \tilde{\boldsymbol{p}'}^\top(\tilde{\boldsymbol{p}}_i - \sum_{r=1}^{l} \text{softmax}(\tilde{\boldsymbol{p}_r}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\tilde{\boldsymbol{p}_r})\tilde{\boldsymbol{p}}_{query}^\top \tilde{\boldsymbol{p}} \\
\leq& -2p_n(t)(1-p_n(t))^2 - 2p_n(t)(1-p_n(t))^2 \cdot \frac{\alpha^2}{K^2} + \frac{1}{l}(\frac{1}{K} - \frac{\alpha}{K})((-1-2p_n(t))K) \\
=& -2p_n(t)(1-p_n(t))^2(1+\frac{\alpha^2}{K^2}) + \frac{1}{l}(1-\alpha)(-1-2p_n(t)).
\end{aligned} \tag{120}$$

We next consider the case where $\tilde{\boldsymbol{p}}$ shares the same positional encoding and the different TRR pattern as $\tilde{\boldsymbol{p}}_{query}$. Then,

$$\begin{aligned}
&(\sum_{i=1}^{l} \text{softmax}(\tilde{\boldsymbol{p}_i}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\boldsymbol{W}_V\tilde{\boldsymbol{p}}_i - \boldsymbol{z}^n)^\top \sum_{i=1}^{l} \text{softmax}(\tilde{\boldsymbol{p}_i}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\boldsymbol{W}_V\tilde{\boldsymbol{p}}_i \\
&\cdot \tilde{\boldsymbol{p}'}^\top(\tilde{\boldsymbol{p}}_i - \sum_{r=1}^{l} \text{softmax}(\tilde{\boldsymbol{p}_r}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\tilde{\boldsymbol{p}_r})\tilde{\boldsymbol{p}}_{query}^\top \tilde{\boldsymbol{p}} \\
\leq& p_n(t)(1-p_n(t))^2 + p_n(t)(1-p_n(t))^2\frac{\alpha^2}{K^2} + \frac{1}{l}(1-\alpha)(-1-2p_n(t)) \\
=& p_n(t)(1-p_n(t))^2(1+\frac{\alpha^2}{K^2}) - \frac{1}{l}(1-\alpha)(1+2p_n(t)).
\end{aligned} \tag{121}$$

Therefore, as long as

$$l \geq \Omega(\alpha^{-1}), \tag{122}$$

we have

$$\begin{aligned}
&\tilde{\boldsymbol{p}'}^\top \eta\frac{1}{B}\sum_{n\in\mathcal{B}_b}\frac{\partial \ell(\Psi;\boldsymbol{P}^n,\boldsymbol{z}^n)}{\partial \boldsymbol{W}}\tilde{\boldsymbol{p}}\\
=&\eta\frac{1}{B}\sum_{n\in\mathcal{B}_b}(F(\Psi;\boldsymbol{P})-\boldsymbol{z}^n)^\top\sum_{i=1}^l \boldsymbol{W}_V\tilde{\boldsymbol{p}}_i\text{softmax}(\tilde{\boldsymbol{p}_i}^\top\boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\tilde{\boldsymbol{p}}^\top(\tilde{\boldsymbol{p}}_i\\
&-\sum_{r=1}^l \text{softmax}(\tilde{\boldsymbol{p}_r}^\top\boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\tilde{\boldsymbol{p}_r})\tilde{\boldsymbol{p}}_{query}^\top\tilde{\boldsymbol{p}}\\
\leq&\eta\frac{1}{B}\sum_{n\in\mathcal{B}_b}(\frac{1}{KM}(-4-2(K-1)(1-p_n(t))(1+\frac{\alpha^2}{K^2}))p_n(t)(1-p_n(t))\\
&+(\frac{1}{K}-\frac{1}{M})p_n(t)(1-p_n(t))^2(1+\frac{\alpha^2}{K^2}))\\
=&\eta\cdot\frac{1}{B}\sum_{n\in\mathcal{B}_b}(\frac{1}{KM}(-4-(3K-2)(1-p_n(t))(1+\frac{\alpha^2}{K^2}))p_n(t)(1-p_n(t))\\
&+\frac{1}{K}p_n(t)(1-p_n(t))^2(1+\frac{\alpha^2}{K^2})),
\end{aligned}\tag{123}$$

and

$$\begin{aligned}
&\tilde{\boldsymbol{p}'}^\top \eta\frac{1}{B}\sum_{n\in\mathcal{B}_b}\frac{\partial \ell(\Psi;\boldsymbol{P}^n,\boldsymbol{z}^n)}{\partial \boldsymbol{W}}\tilde{\boldsymbol{p}}\\
\geq&\eta\frac{1}{B}\sum_{n\in\mathcal{B}_b}(\frac{1}{KM}(-4-(3K-2)(1-p_n(t))(1+\frac{\alpha^2}{K^2}))p_n(t)(1-p_n(t))\\
&+\frac{1}{K}p_n(t)(1-p_n(t))^2(1+\frac{\alpha^2}{K^2})+\frac{1}{K}\cdot(1-p_n(t))^2(-p_n(t)+(1-p_n(t))\frac{\alpha^2}{K^2}))\\
=&\eta\cdot\frac{1}{B}\sum_{n\in\mathcal{B}_b}(\frac{1}{KM}(-4-(3K-2)(1-p_n(t))(1+\frac{\alpha^2}{K^2}))p_n(t)(1-p_n(t))+\frac{\alpha^2}{K^3}(1-p_n(t))^2).
\end{aligned}\tag{124}$$

We next consider the case where $\tilde{\boldsymbol{p}}'$ shares a different TRR pattern and the same positional encoding as $\tilde{\boldsymbol{p}}$. Let $\tilde{\boldsymbol{p}}$ share the same TRR pattern and the positional encoding as $\tilde{\boldsymbol{p}}_{query}$. If $\tilde{\boldsymbol{p}}'$ and $\tilde{\boldsymbol{p}}_i$ share the same TRR pattern, label pattern, and positional encoding,

$$\begin{aligned}
2(3-p_n(t))\geq\tilde{\boldsymbol{p}'}^\top(\tilde{\boldsymbol{p}}_i-\sum_{r=1}^l\text{softmax}(\tilde{\boldsymbol{p}_r}^\top\boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\tilde{\boldsymbol{p}_r})\tilde{\boldsymbol{p}}_{query}^\top\tilde{\boldsymbol{p}}\geq&2\cdot(3-p_n(t)-(1-p_n(t)))\\
=&4.
\end{aligned}\tag{125}$$

When $\tilde{\boldsymbol{p}}'$ and $\tilde{\boldsymbol{p}}_i$ only share the same positional encoding or the same TRR pattern,

$$2(1-p_n(t))\geq\tilde{\boldsymbol{p}'}^\top(\tilde{\boldsymbol{p}}_i-\sum_{r=1}^l\text{softmax}(\tilde{\boldsymbol{p}_r}^\top\boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\tilde{\boldsymbol{p}_r})\tilde{\boldsymbol{p}}_{query}^\top\tilde{\boldsymbol{p}}\geq 0.\tag{126}$$

When $\tilde{\boldsymbol{p}}'$ and $\tilde{\boldsymbol{p}}_i$ only share both different positional encodings and TRR patterns,

$$-2p_n(t)\geq\tilde{\boldsymbol{p}'}^\top(\tilde{\boldsymbol{p}}_i-\sum_{r=1}^l\text{softmax}(\tilde{\boldsymbol{p}_r}^\top\boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\tilde{\boldsymbol{p}_r})\tilde{\boldsymbol{p}}_{query}^\top\tilde{\boldsymbol{p}}\geq -2.\tag{127}$$

Then, we consider the case where $\tilde{\boldsymbol{p}}$ only shares the same TRR pattern as $\tilde{\boldsymbol{p}}_{query}$. If $\tilde{\boldsymbol{p}}'$ and $\tilde{\boldsymbol{p}}_i$ share the same TRR pattern, label pattern, and the positional encoding,

$$3\geq\tilde{\boldsymbol{p}'}^\top(\tilde{\boldsymbol{p}}_i-\sum_{r=1}^l\text{softmax}(\tilde{\boldsymbol{p}_r}^\top\boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\tilde{\boldsymbol{p}_r})\tilde{\boldsymbol{p}}_{query}^\top\tilde{\boldsymbol{p}}\geq 1\cdot(3-(1-p_n(t)))=2+p_n(t).\tag{128}$$

When $\tilde{\boldsymbol{p}}'$ and $\tilde{\boldsymbol{p}}_i$ only share the same positional encoding or the same TRR pattern,

$$1 \geq \tilde{\boldsymbol{p}'}^\top(\tilde{\boldsymbol{p}}_i - \sum_{r=1}^{l} \text{softmax}(\tilde{\boldsymbol{p}_r}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\tilde{\boldsymbol{p}_r})\tilde{\boldsymbol{p}}_{query}^\top\tilde{\boldsymbol{p}} \geq p_n(t). \tag{129}$$

When $\tilde{\boldsymbol{p}}'$ and $\tilde{\boldsymbol{p}}_i$ only share both different positional encodings and TRR patterns,

$$0 \geq \tilde{\boldsymbol{p}'}^\top(\tilde{\boldsymbol{p}}_i - \sum_{r=1}^{l} \text{softmax}(\tilde{\boldsymbol{p}_r}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\tilde{\boldsymbol{p}_r})\tilde{\boldsymbol{p}}_{query}^\top\tilde{\boldsymbol{p}} \geq -1 + p_n(t). \tag{130}$$

Next, we consider the case where $\tilde{\boldsymbol{p}}$ only shares the same positional encoding as $\tilde{\boldsymbol{p}}_{query}$. If $\tilde{\boldsymbol{p}}'$ and $\tilde{\boldsymbol{p}}_i$ share the same TRR pattern, label pattern, and the positional encoding,

$$3 - p_n(t) \geq \tilde{\boldsymbol{p}'}^\top(\tilde{\boldsymbol{p}}_i - \sum_{r=1}^{l} \text{softmax}(\tilde{\boldsymbol{p}_r}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\tilde{\boldsymbol{p}_r})\tilde{\boldsymbol{p}}_{query}^\top\tilde{\boldsymbol{p}} \geq 1 \cdot (3 - p_n(t) - (1 - p_n(t))) = 2. \tag{131}$$

When $\tilde{\boldsymbol{p}}'$ and $\tilde{\boldsymbol{p}}_i$ only share the same positional encoding or the same TRR pattern,

$$1 - p_n(t) \geq \tilde{\boldsymbol{p}'}^\top(\tilde{\boldsymbol{p}}_i - \sum_{r=1}^{l} \text{softmax}(\tilde{\boldsymbol{p}_r}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\tilde{\boldsymbol{p}_r})\tilde{\boldsymbol{p}}_{query}^\top\tilde{\boldsymbol{p}} \geq 0. \tag{132}$$

When $\tilde{\boldsymbol{p}}'$ and $\tilde{\boldsymbol{p}}_i$ only share both different positional encodings and TRR patterns,

$$-p_n(t) \geq \tilde{\boldsymbol{p}'}^\top(\tilde{\boldsymbol{p}}_i - \sum_{r=1}^{l} \text{softmax}(\tilde{\boldsymbol{p}_r}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\tilde{\boldsymbol{p}_r})\tilde{\boldsymbol{p}}_{query}^\top\tilde{\boldsymbol{p}} \geq -1. \tag{133}$$

Then, when $l \geq \Omega(\alpha^{-1})$, and when $\tilde{\boldsymbol{p}}$ shares the same TRR pattern and the positional encoding as $\tilde{\boldsymbol{p}}_{query}$, by (105),

$$\begin{aligned}
&(\sum_{i=1}^{l} \text{softmax}(\tilde{\boldsymbol{p}_i}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\boldsymbol{W}_V\tilde{\boldsymbol{p}}_i - \boldsymbol{z}^n)^\top \sum_{i=1}^{l} \text{softmax}(\tilde{\boldsymbol{p}_i}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\boldsymbol{W}_V\tilde{\boldsymbol{p}}_i \\
&\cdot \tilde{\boldsymbol{p}'}^\top(\tilde{\boldsymbol{p}}_i - \sum_{r=1}^{l} \text{softmax}(\tilde{\boldsymbol{p}_r}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\tilde{\boldsymbol{p}_r})\tilde{\boldsymbol{p}}_{query}^\top\tilde{\boldsymbol{p}} \\
\leq & 0 - 2(1 - p_n(t))^2 \frac{\alpha^2}{K^2} + \frac{1}{l}(\frac{1}{K} - \frac{\alpha}{K})(-2(K-1)).
\end{aligned} \tag{134}$$

We next consider the case where $\tilde{\boldsymbol{p}}$ shares the same TRR pattern and the different positional encoding as $\tilde{\boldsymbol{p}}_{query}$. Then,

$$\begin{aligned}
&(\sum_{i=1}^{l} \text{softmax}(\tilde{\boldsymbol{p}_i}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\boldsymbol{W}_V\tilde{\boldsymbol{p}}_i - \boldsymbol{z}^n)^\top \sum_{i=1}^{l} \text{softmax}(\tilde{\boldsymbol{p}_i}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\boldsymbol{W}_V\tilde{\boldsymbol{p}}_i \\
&\cdot \tilde{\boldsymbol{p}'}^\top(\tilde{\boldsymbol{p}}_i - \sum_{r=1}^{l} \text{softmax}(\tilde{\boldsymbol{p}_r}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\tilde{\boldsymbol{p}_r})\tilde{\boldsymbol{p}}_{query}^\top\tilde{\boldsymbol{p}} \\
\leq & -p_n(t)(1 - p_n(t))(-1 + p_n(t)) + p_n(t)(1 - p_n(t))^2 \cdot \frac{\alpha^2}{K^2} + \frac{1}{l}(\frac{1}{K} - \frac{\alpha}{K})K(-1 + p_n(t)) \\
= & p_n(t)(1 - p_n(t))^2(\frac{\alpha^2}{K^2} + 1) + \frac{1}{l}(1 - \alpha)(-1 + p_n(t)).
\end{aligned} \tag{135}$$

We next consider the case where $\tilde{\boldsymbol{p}}$ shares the same positional encoding and the different TRR pattern as $\tilde{\boldsymbol{p}}_{query}$. Then,

$$\begin{aligned}
&(\sum_{i=1}^l \text{softmax}(\tilde{\boldsymbol{p}_i}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\boldsymbol{W}_V\tilde{\boldsymbol{p}}_i-\boldsymbol{z}^n)^\top \sum_{i=1}^l \text{softmax}(\tilde{\boldsymbol{p}_i}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\boldsymbol{W}_V\tilde{\boldsymbol{p}}_i\\
&\cdot \tilde{\boldsymbol{p}'}^\top (\tilde{\boldsymbol{p}}_i-\sum_{r=1}^l \text{softmax}(\tilde{\boldsymbol{p}_r}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\tilde{\boldsymbol{p}_r})\tilde{\boldsymbol{p}}_{query}^\top\tilde{\boldsymbol{p}}\\
\leq &-(1-p_n(t))^2\frac{\alpha^2}{K^2}-0+\frac{1}{l}(\frac{1}{K}-\frac{\alpha}{K})(-K+1)\\
=&-(1-p_n(t))^2\frac{\alpha^2}{K^2}-\frac{K-1}{Kl}(1-\alpha).
\end{aligned}\tag{136}$$

Therefore, as long as

$$l\geq \Omega(\alpha^{-1}),\tag{137}$$

we have

$$\begin{aligned}
&\tilde{\boldsymbol{p}'}^\top \eta\frac{1}{B}\sum_{n\in\mathcal{B}_b}\frac{\partial \ell(\Psi;\boldsymbol{P}^n,\boldsymbol{z}^n)}{\partial \boldsymbol{W}}\boldsymbol{p}\\
=&\eta\frac{1}{B}\sum_{n\in\mathcal{B}_b}(F(\Psi;\boldsymbol{P})-\boldsymbol{z}^n)^\top\sum_{i=1}^l \boldsymbol{W}_V\tilde{\boldsymbol{p}}_i\text{softmax}(\tilde{\boldsymbol{p}_i}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\\
&\cdot\tilde{\boldsymbol{p}}^\top(\tilde{\boldsymbol{p}}_i-\sum_{r=1}^l \text{softmax}(\tilde{\boldsymbol{p}_r}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\tilde{\boldsymbol{p}_r})\tilde{\boldsymbol{p}}_{query}^\top\tilde{\boldsymbol{p}}\\
\leq&\eta\frac{1}{B}\sum_{n\in\mathcal{B}_b}(\frac{1}{KM}(-\frac{\alpha^2}{K^2}+(K-1)(1+\frac{\alpha^2}{K^2})p_n(t))(1-p_n(t))^2-(\frac{1}{K}\\
&-\frac{1}{M})(1-p_n(t))^2\frac{\alpha^2}{K^2}))\\
=&\eta\cdot\frac{1}{B}\sum_{n\in\mathcal{B}_b}(\frac{1}{KM}(-\frac{\alpha^2}{K^2}+(K-1+\frac{(2K-1)\alpha^2}{K^2})p_n(t))(1-p_n(t))^2-(1-p_n(t))^2\frac{\alpha^2}{K^3}).
\end{aligned}\tag{138}$$

and

$$\begin{aligned}
&\tilde{\boldsymbol{p}'}^\top \eta\frac{1}{B}\sum_{n\in\mathcal{B}_b}\frac{\partial \ell(\Psi;\boldsymbol{P}^n,\boldsymbol{z}^n)}{\partial \boldsymbol{W}}\boldsymbol{p}\\
\geq&\eta\cdot\frac{1}{B}\sum_{n\in\mathcal{B}_b}(\frac{1}{KM}(-\frac{\alpha^2}{K^2}+(K-1+\frac{(2K-1)\alpha^2}{K^2})p_n(t))(1-p_n(t))^2-(1-p_n(t))^2\frac{\alpha^2}{K^3}\\
&+\frac{1}{K}\cdot(1-p_n(t))^2(-p_n(t)+(1-p_n(t))\frac{\alpha^2}{K^2})).
\end{aligned}\tag{139}$$

We next consider the case where $\tilde{\boldsymbol{p}}'$ shares a different TRR pattern and a different positional encoding as $\tilde{\boldsymbol{p}}$. Let $\tilde{\boldsymbol{p}}$ share the same TRR pattern and the positional encoding as $\tilde{\boldsymbol{p}}_{query}$. If $\tilde{\boldsymbol{p}}'$ and $\tilde{\boldsymbol{p}}_i$ share the same TRR pattern, label pattern, and the positional encoding,

$$6\geq \tilde{\boldsymbol{p}'}^\top(\tilde{\boldsymbol{p}}_i-\sum_{r=1}^l \text{softmax}(\tilde{\boldsymbol{p}_r}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\tilde{\boldsymbol{p}_r})\tilde{\boldsymbol{p}}_{query}^\top\tilde{\boldsymbol{p}}\geq 2\cdot(3-(1-p_n(t)))=4+2p_n(t).\tag{140}$$

When $\tilde{\boldsymbol{p}}'$ and $\tilde{\boldsymbol{p}}_i$ only share the same positional encoding or the same TRR pattern,

$$2\geq \tilde{\boldsymbol{p}'}^\top(\tilde{\boldsymbol{p}}_i-\sum_{r=1}^l \text{softmax}(\tilde{\boldsymbol{p}_r}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\tilde{\boldsymbol{p}_r})\tilde{\boldsymbol{p}}_{query}^\top\tilde{\boldsymbol{p}}\geq 2p_n(t).\tag{141}$$

When $\tilde{\boldsymbol{p}}'$ and $\tilde{\boldsymbol{p}}_i$ only share both different positional encodings and TRR patterns,

$$0 \geq \tilde{\boldsymbol{p}'}^\top(\tilde{\boldsymbol{p}}_i - \sum_{r=1}^{l} \text{softmax}(\tilde{\boldsymbol{p}_r}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\tilde{\boldsymbol{p}}_r)\tilde{\boldsymbol{p}}_{query}^\top \tilde{\boldsymbol{p}} \geq -2 + 2p_n(t). \tag{142}$$

Then, we consider the case where $\tilde{\boldsymbol{p}}$ only shares the same TRR pattern as $\tilde{\boldsymbol{p}}_{query}$. If $\tilde{\boldsymbol{p}}'$ and $\tilde{\boldsymbol{p}}_i$ share the same TRR pattern, label pattern, and the positional encoding,

$$3 \geq \tilde{\boldsymbol{p}'}^\top(\tilde{\boldsymbol{p}}_i - \sum_{r=1}^{l} \text{softmax}(\tilde{\boldsymbol{p}_r}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\tilde{\boldsymbol{p}}_r)\tilde{\boldsymbol{p}}_{query}^\top \tilde{\boldsymbol{p}} \geq 1 \cdot (3 - p_n(t) - (1 - p_n(t))) = 2. \tag{143}$$

When $\tilde{\boldsymbol{p}}'$ and $\tilde{\boldsymbol{p}}_i$ only share the same positional encoding or the same TRR pattern,

$$1 \geq \tilde{\boldsymbol{p}'}^\top(\tilde{\boldsymbol{p}}_i - \sum_{r=1}^{l} \text{softmax}(\tilde{\boldsymbol{p}_r}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\tilde{\boldsymbol{p}}_r)\tilde{\boldsymbol{p}}_{query}^\top \tilde{\boldsymbol{p}} \geq 0. \tag{144}$$

When $\tilde{\boldsymbol{p}}'$ and $\tilde{\boldsymbol{p}}_i$ only share both different positional encodings and TRR patterns,

$$0 \geq \tilde{\boldsymbol{p}'}^\top(\tilde{\boldsymbol{p}}_i - \sum_{r=1}^{l} \text{softmax}(\tilde{\boldsymbol{p}_r}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\tilde{\boldsymbol{p}}_r)\tilde{\boldsymbol{p}}_{query}^\top \tilde{\boldsymbol{p}} \geq -1. \tag{145}$$

Next, we consider the case where $\tilde{\boldsymbol{p}}$ only shares the same positional encoding as $\tilde{\boldsymbol{p}}_{query}$. If $\tilde{\boldsymbol{p}}'$ and $\tilde{\boldsymbol{p}}_i$ share the same TRR pattern, label pattern, and the positional encoding,

$$3 \geq \tilde{\boldsymbol{p}'}^\top(\tilde{\boldsymbol{p}}_i - \sum_{r=1}^{l} \text{softmax}(\tilde{\boldsymbol{p}_r}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\tilde{\boldsymbol{p}}_r)\tilde{\boldsymbol{p}}_{query}^\top \tilde{\boldsymbol{p}} \geq 1 \cdot (3 - (1 - p_n(t))) = 2 + p_n(t). \tag{146}$$

When $\tilde{\boldsymbol{p}}'$ and $\tilde{\boldsymbol{p}}_i$ only share the same positional encoding or the same TRR pattern,

$$1 \geq \tilde{\boldsymbol{p}'}^\top(\tilde{\boldsymbol{p}}_i - \sum_{r=1}^{l} \text{softmax}(\tilde{\boldsymbol{p}_r}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\tilde{\boldsymbol{p}_r})\tilde{\boldsymbol{p}}_{query}^\top \tilde{\boldsymbol{p}} \geq p_n(t). \tag{147}$$

When $\tilde{\boldsymbol{p}}'$ and $\tilde{\boldsymbol{p}}_i$ only share both different positional encodings and TRR patterns,

$$0 \geq \tilde{\boldsymbol{p}'}^\top(\tilde{\boldsymbol{p}}_i - \sum_{r=1}^{l} \text{softmax}(\tilde{\boldsymbol{p}_r}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\tilde{\boldsymbol{p}_r})\tilde{\boldsymbol{p}}_{query}^\top \tilde{\boldsymbol{p}} \geq -1 + p_n(t). \tag{148}$$

Then, when $l \geq \Omega(\alpha^{-1})$, and when $\tilde{\boldsymbol{p}}$ shares the same TRR pattern and the positional encoding as $\tilde{\boldsymbol{p}}_{query}$,

$$\begin{aligned}
&(\sum_{i=1}^{l} \text{softmax}(\tilde{\boldsymbol{p}_i}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\boldsymbol{W}_V\tilde{\boldsymbol{p}}_i - \boldsymbol{z}^n)^\top \sum_{i=1}^{l} \text{softmax}(\tilde{\boldsymbol{p}_i}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\boldsymbol{W}_V\tilde{\boldsymbol{p}}_i \\
&\cdot \tilde{\boldsymbol{p}'}^\top(\tilde{\boldsymbol{p}}_i - \sum_{r=1}^{l} \text{softmax}(\tilde{\boldsymbol{p}_r}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\tilde{\boldsymbol{p}_r})\tilde{\boldsymbol{p}}_{query}^\top \tilde{\boldsymbol{p}} \\
\leq& - p_n(t)(1 - p_n(t))(-2 + 2p_n(t)) + (1 - p_n(t))^2 \frac{\alpha^2}{K^2} \cdot 2p_n(t) + \frac{1}{l}(1 - \alpha)(-2 + 2p_n(t)).
\end{aligned} \tag{149}$$

We next consider the case where $\tilde{\boldsymbol{p}}$ shares the same TRR pattern and the different positional encoding as $\tilde{\boldsymbol{p}}_{query}$. Then,

$$\begin{aligned}
&(\sum_{i=1}^{l} \text{softmax}(\tilde{\boldsymbol{p}_i}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\boldsymbol{W}_V\tilde{\boldsymbol{p}}_i - \boldsymbol{z}^n)^\top \sum_{i=1}^{l} \text{softmax}(\tilde{\boldsymbol{p}_i}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\boldsymbol{W}_V\tilde{\boldsymbol{p}}_i \\
&\cdot \tilde{\boldsymbol{p}'}^\top(\tilde{\boldsymbol{p}}_i - \sum_{r=1}^{l} \text{softmax}(\tilde{\boldsymbol{p}_r}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\tilde{\boldsymbol{p}}_r)\tilde{\boldsymbol{p}}_{query}^\top \tilde{\boldsymbol{p}} \\
\leq& 0 + p_n(t)(1 - p_n(t))^2 \cdot \frac{\alpha^2}{K^2} \cdot (-1) + \frac{1}{l}(\frac{1}{K} - \frac{\alpha}{K})(-K) \\
=& - p_n(t)(1 - p_n(t))^2 \frac{\alpha^2}{K^2} + \frac{1}{l}(1 - \alpha)(-1).
\end{aligned} \tag{150}$$

We next consider the case where $\tilde{\boldsymbol{p}}$ shares the same positional encoding and the different TRR pattern as $\tilde{\boldsymbol{p}}_{query}$. Then,

$$\begin{aligned}
&(\sum_{i=1}^l \text{softmax}(\tilde{\boldsymbol{p}_i}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\boldsymbol{W}_V\tilde{\boldsymbol{p}}_i-\boldsymbol{z}^n)^\top\sum_{i=1}^l \text{softmax}(\tilde{\boldsymbol{p}_i}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\boldsymbol{W}_V\tilde{\boldsymbol{p}}_i\\
&\cdot\tilde{\boldsymbol{p}'}^\top(\tilde{\boldsymbol{p}}_i-\sum_{r=1}^l \text{softmax}(\tilde{\boldsymbol{p}_r}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\tilde{\boldsymbol{p}_r})\tilde{\boldsymbol{p}}_{query}^\top\tilde{\boldsymbol{p}}\\
\leq&-(1-p_n(t))p_n(t)(-1+p_n(t))+p_n(t)(1-p_n(t))^2\frac{\alpha^2}{K^2}+\frac{1}{l}(\frac{1}{K}-\frac{\alpha}{K})(-1+p_n(t))K\\
=&(1-p_n(t))^2p_n(t)(1+\frac{\alpha^2}{K^2})+\frac{1}{l}(1-\alpha)(-1+p_n(t)).
\end{aligned}\tag{151}$$

Therefore, as long as

$$l\geq\Omega(\alpha^{-1}),\tag{152}$$

we have

$$\begin{aligned}
&\tilde{\boldsymbol{p}'}^\top\eta\frac{1}{B}\sum_{n\in\mathcal{B}_b}\frac{\partial\ell(\Psi;\boldsymbol{P}^n,\boldsymbol{z}^n)}{\partial\boldsymbol{W}}\boldsymbol{p}\\
=&\eta\frac{1}{B}\sum_{n\in\mathcal{B}_b}(F(\Psi;\boldsymbol{P})-\boldsymbol{z}^n)^\top\sum_{i=1}^l\boldsymbol{W}_V\tilde{\boldsymbol{p}_i}\text{softmax}(\tilde{\boldsymbol{p}_i}^\top\boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\\
&\cdot\tilde{\boldsymbol{p}}^\top(\tilde{\boldsymbol{p}}_i-\sum_{r=1}^l\text{softmax}(\tilde{\boldsymbol{p}_r}^\top\boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\tilde{\boldsymbol{p}}_r)\tilde{\boldsymbol{p}}_{query}^\top\tilde{\boldsymbol{p}}\\
\leq&\eta\frac{1}{B}\sum_{n\in\mathcal{B}_b}(\frac{1}{KM}(-p_n(t)(1-p_n(t))(-2+2p_n(t))+(3-K)(1-p_n(t))^2\frac{\alpha^2}{K^2}\cdot p_n(t))\\
&+(\frac{1}{K}-\frac{1}{M})(1-p_n(t))^2p_n(t)(1+\frac{\alpha^2}{K^2}))\\
=&\eta\cdot\frac{1}{B}\sum_{n\in\mathcal{B}_b}(\frac{1}{KM}p_n(t)(1-p_n(t))^2(2-K+\frac{(2-K)\alpha^2}{K^2})\\
&+(1-p_n(t))^2p_n(t)(1+\frac{\alpha^2}{K^2})\cdot\frac{1}{K}),
\end{aligned}\tag{153}$$

and

$$\begin{aligned}
&\tilde{\boldsymbol{p}'}^\top\eta\frac{1}{B}\sum_{n\in\mathcal{B}_b}\frac{\partial\ell(\Psi;\boldsymbol{P}^n,\boldsymbol{z}^n)}{\partial\boldsymbol{W}}\boldsymbol{p}\\
\geq&\eta\cdot\frac{1}{B}\sum_{n\in\mathcal{B}_b}(\frac{1}{KM}p_n(t)(1-p_n(t))^2(1+\frac{(2-K)\alpha^2}{K^2})+(1-p_n(t))^2p_n(t)(1+\frac{\alpha^2}{K^2})\cdot\frac{1}{K}\\
&+\frac{1}{K}\cdot(1-p_n(t))^2(-p_n(t)+(1-p_n(t))\frac{\alpha^2}{K^2}))\\
=&\eta\cdot\frac{1}{B}\sum_{n\in\mathcal{B}_b}(\frac{1}{KM}p_n(t)(1-p_n(t))^2(1+\frac{(2-K)\alpha^2}{K^2})+(1-p_n(t))^2\cdot\frac{\alpha^2}{K^3}).
\end{aligned}\tag{154}$$

□

H.2. Proof of Lemma F.6

Proof. We can derive that when $1-p_n(t)\geq\Omega(1)$, $\tilde{\boldsymbol{p}'}^\top\boldsymbol{W}^{(t)}\tilde{\boldsymbol{p}}$ increases if $\tilde{\boldsymbol{p}}$ and $\tilde{\boldsymbol{p}}'$ share the same positional encoding. Otherwise, $\tilde{\boldsymbol{p}'}^\top\boldsymbol{W}^{(t)}\tilde{\boldsymbol{p}}$ decreases. We know that $p_n(t)\geq\frac{\alpha}{2}$. Combining the results in Lemma F.5, we can derive that when $t\geq 1$,

$$\boldsymbol{W}^{(t+1)}=\boldsymbol{W}^{(t)}-\eta\frac{1}{B}\sum_{n\in\mathcal{B}_b}\frac{\partial\ell(\Psi;\boldsymbol{P}^n,\boldsymbol{z}^n)}{\partial\boldsymbol{W}^{(t)}}.\tag{155}$$

Then, for $\tilde{\boldsymbol{p}}_i{}^n$ that share the same TRR pattern and the same positional encoding of $\tilde{\boldsymbol{p}}_{query}^n$,

$$\begin{aligned}\frac{p_n(t+1)}{|\mathcal{S}_1^n|}=&\text{softmax}(\boldsymbol{p}_i^{n\top}\boldsymbol{W}^{(t+1)}\tilde{\boldsymbol{p}}_{query}^n)\\ \geq&\frac{1}{l}\cdot\frac{1}{\frac{\alpha}{K}+\frac{(K-1)\alpha}{K}\cdot e^{-s_1}+(\frac{1}{K}-\frac{\alpha}{K})((K-1)e^{-s_2}+e^{-s_3})},\end{aligned}\tag{156}$$

where

$$s_1\geq\eta\sum_{b=0}^t((1-p_n(b))^2\frac{\alpha^2}{K^3}+\frac{\alpha^2}{K^3}(1-p_n(b))^2)=\eta\sum_{b=0}^t(1-p_n(b))^2\frac{2\alpha^2}{K^3},\tag{157}$$

$$s_2\geq\sum_{b=0}^t(1-p_n(b))^2\cdot\frac{2\eta\alpha^2}{K^3},\tag{158}$$

$$\begin{aligned}s_3\geq&-\frac{\eta}{KM}\sum_{b=0}^t(1-p_n(b)^2(-4p_n(b)(1+\frac{\alpha^2}{K^2})+\frac{\alpha^2}{K}(1+\frac{2(K-1)}{K})+\frac{\alpha^2}{K^2}\\ &-(K-1+\frac{2K-1}{K^2}\alpha^2)p_n(b)))\\ \geq&\frac{\eta}{KM}\sum_{b=0}^t(1-p_n(b))^2(p_n(b)(3+\frac{\alpha^2}{K^2})(4+\frac{2K-1}{K^2})),\end{aligned}\tag{159}$$

where the last step is by $Kp_n(b)\geq 4\alpha^2/K^2$ when $p_n(b)\geq\alpha/K$. For $\tilde{\boldsymbol{p}}_i{}^n$ that share the same TRR pattern and a different positional encoding of $\tilde{\boldsymbol{p}}_{query}^n$,

$$\text{softmax}(\tilde{\boldsymbol{p}_i^n}^\top\boldsymbol{W}^{(t+1)}\tilde{\boldsymbol{p}}_{query}^n)=\frac{1}{l}\cdot\frac{1}{\frac{\alpha}{K}e^{s_1}+\frac{(K-1)\alpha}{K}+(\frac{1}{K}-\frac{\alpha}{K})((K-1)e^{-s_4}+e^{s_5})},\tag{160}$$

where

$$\begin{aligned}s_4\geq&-\sum_{b=0}^t\frac{\eta}{M}((-4-(3K-2)(1-p_n(b))(1+\frac{\alpha^2}{K^2}))p_n(b)(1-p_n(b))\\ &-(2-K)(1+\frac{\alpha^2}{K^2})p_n(b)(1-p_n(b))^2)\\ =&\sum_{b=0}^t\frac{\eta}{M}(4+2K(1-p_n(b))(1+\frac{\alpha^2}{K^2}))p_n(b)(1-p_n(b)),\end{aligned}\tag{161}$$

$$s_5\geq\sum_{b=0}^t(1-p_n(b))^2\cdot\frac{2\eta\alpha^2}{K^3}.\tag{162}$$

When $M\geq\Omega(K^4\alpha^{-1})$ and $t\geq\Omega(\eta^{-1}K^3\log K\alpha^{-2})$,

$$(K-1)e^{-s_4}+e^{s_5}>K.\tag{163}$$

If $M\geq\Omega(K^4\alpha^{-1})$ and $t\leq O(\eta^{-1}K^3\log K\alpha^{-2})$, we cannot ensure

$$(K-1)e^{-s_4}+e^{s_5}>K.\tag{164}$$

For $\tilde{\boldsymbol{p}_i^n}$ that share a different TRR pattern and the same positional encoding of $\tilde{\boldsymbol{p}}_{query}^n$,

$$\text{softmax}(\tilde{\boldsymbol{p}_i^n}^\top\boldsymbol{W}^{(t+1)}\tilde{\boldsymbol{p}}_{query}^n)=\frac{1}{l}\cdot\frac{1}{\frac{\alpha}{K}e^{s_3}+\frac{\alpha}{K}\cdot e^{-s_4}+(\frac{1}{K}-\frac{\alpha}{K})(1+(K-1)e^{-s_6})},\tag{165}$$

where

$$s_6\geq\eta\sum_{b=0}^t\frac{2\alpha^2}{K^3}(1-p_n(b))^2.\tag{166}$$

For $\tilde{\boldsymbol{p}_i^n}$ that share a different TRR pattern and a different positional encoding of $\tilde{\boldsymbol{p}}_{query}^n$,

$$\text{softmax}(\tilde{\boldsymbol{p}_i^n}^\top \boldsymbol{W}^{(t+1)}\tilde{\boldsymbol{p}}_{query}^n) = \frac{1}{l}\cdot\frac{1}{\frac{\alpha}{K}e^{s_2}+(\frac{1}{K}-\frac{\alpha}{K})(K-1+e^{s_6})+\frac{\alpha}{K}e^{s_4}}. \tag{167}$$

Note that when $t \lesssim \eta^{-1}\alpha^{-2}K^3$, for $\boldsymbol{p}_{query}^n$ in the k-th step, we have

$$\sum_{i\in\mathcal{S}_{[K]\setminus\{k\}}} \text{softmax}(\tilde{\boldsymbol{p}_i^n}^\top \boldsymbol{W}^{(t+1)}\tilde{\boldsymbol{p}}_{query}^n) \geq \Omega(1), \tag{168}$$

for $\tilde{\boldsymbol{p}_i^n}$ that share a different positional encoding from $\tilde{\boldsymbol{p}}_{query}^n$. To make the total softmax values on contexts that share a different positional encoding and a different TRR pattern from the query smaller than ϵ, we need

$$s_1, s_2, s_6 \gtrsim \log\frac{K}{\epsilon}. \tag{169}$$

When t further increases to be larger than $\Omega(\eta^{-1}\alpha^{-2}K^3\log\frac{K}{\epsilon})$, we also have that the total softmax values on contexts that share a different positional encoding and the same TRR pattern from the query smaller than ϵ. Therefore,

$$t \gtrsim T_1 := \eta^{-1}\alpha^{-2}K^3\log\frac{K}{\epsilon}. \tag{170}$$

$\square$

H.3. Proof of Lemma F.7

Proof. We consider the case when $t \geq T_1$ given Lemma F.6. When $l \geq \Omega(\alpha^{-1})$, and when $\tilde{\boldsymbol{p}}$ shares the same TRR pattern and the positional encoding as $\tilde{\boldsymbol{p}}_{query}$,

$$\begin{aligned}
&(\sum_{i=1}^l \text{softmax}(\tilde{\boldsymbol{p}_i}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\boldsymbol{W}_V\tilde{\boldsymbol{p}}_i - \boldsymbol{z}^n)^\top \sum_{i=1}^l \text{softmax}(\tilde{\boldsymbol{p}_i}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\boldsymbol{W}_V\tilde{\boldsymbol{p}}_i\\
&\cdot \tilde{\boldsymbol{p}}^\top(\tilde{\boldsymbol{p}}_i - \sum_{r=1}^l \text{softmax}(\tilde{\boldsymbol{p}_r}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\tilde{\boldsymbol{p}_r})\tilde{\boldsymbol{p}}_{query}^\top\tilde{\boldsymbol{p}}\\
\leq& -4p_n(t)(1-p_n(t))^2+\epsilon\\
\lesssim& -4p_n(t)(1-p_n(t))^2.
\end{aligned} \tag{171}$$

We next consider the case where $\tilde{\boldsymbol{p}}$ shares the same TRR pattern and the different positional encoding as $\tilde{\boldsymbol{p}}_{query}$. Then,

$$\begin{aligned}
&(\sum_{i=1}^l \text{softmax}(\tilde{\boldsymbol{p}_i}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\boldsymbol{W}_V\tilde{\boldsymbol{p}}_i - \boldsymbol{z}^n)^\top \sum_{i=1}^l \text{softmax}(\tilde{\boldsymbol{p}_i}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\boldsymbol{W}_V\tilde{\boldsymbol{p}}_i\\
&\cdot \tilde{\boldsymbol{p}}^\top(\tilde{\boldsymbol{p}}_i - \sum_{r=1}^l \text{softmax}(\tilde{\boldsymbol{p}_r}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\tilde{\boldsymbol{p}_r})\tilde{\boldsymbol{p}}_{query}^\top\tilde{\boldsymbol{p}}\\
\lesssim& -0\cdot p_n(t)(1-p_n(t))+\epsilon\\
\lesssim&\epsilon.
\end{aligned} \tag{172}$$

We next consider the case where $\tilde{\boldsymbol{p}}$ shares the same positional encoding and the different TRR pattern as $\tilde{\boldsymbol{p}}_{query}$. Then,

$$\begin{aligned}
&(\sum_{i=1}^l \text{softmax}(\tilde{\boldsymbol{p}_i}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\boldsymbol{W}_V\tilde{\boldsymbol{p}}_i - \boldsymbol{z}^n)^\top \sum_{i=1}^l \text{softmax}(\tilde{\boldsymbol{p}_i}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\boldsymbol{W}_V\tilde{\boldsymbol{p}}_i\\
&\cdot \tilde{\boldsymbol{p}}^\top(\tilde{\boldsymbol{p}}_i - \sum_{r=1}^l \text{softmax}(\tilde{\boldsymbol{p}_r}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\tilde{\boldsymbol{p}_r})\tilde{\boldsymbol{p}}_{query}^\top\tilde{\boldsymbol{p}}\\
\lesssim&\epsilon.
\end{aligned} \tag{173}$$

Therefore,

$$\begin{aligned}
&\tilde{\boldsymbol{p}}^\top \eta \frac{1}{B}\sum_{n\in\mathcal{B}_b}\frac{\partial \ell(\Psi;\boldsymbol{P}^n,\boldsymbol{z}^n)}{\partial \boldsymbol{W}}\boldsymbol{p}\\
=&\eta\frac{1}{B}\sum_{n\in\mathcal{B}_b}(F(\Psi;\boldsymbol{P})-\boldsymbol{z}^n)^\top\sum_{i=1}^l \boldsymbol{W}_V\tilde{\boldsymbol{p}}_i\text{softmax}(\tilde{\boldsymbol{p}_i}^\top\boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\\
&\cdot\tilde{\boldsymbol{p}}^\top(\tilde{\boldsymbol{p}}_i-\sum_{r=1}^l\text{softmax}(\tilde{\boldsymbol{p}_r}^\top\boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\tilde{\boldsymbol{p}_r})\tilde{\boldsymbol{p}}_{query}^\top\tilde{\boldsymbol{p}}\\
\lesssim&\eta\frac{1}{B}\sum_{n\in\mathcal{B}_b}(\frac{1}{2M}(-4p_n(t)(1-p_n(t))^2)+(\frac{1}{2}-\frac{1}{M})\cdot\epsilon\\
=&-\eta\cdot\frac{1}{2M}\cdot\frac{1}{B}\sum_{n\in\mathcal{B}_b}4p_n(t)(1-p_n(t))^2.
\end{aligned}\tag{174}$$

We then discuss if $\tilde{\boldsymbol{p}}$ and $\tilde{\boldsymbol{p}}'$ only share the same TRR pattern. When $l\geq\Omega(\alpha^{-1})$, and when $\tilde{\boldsymbol{p}}$ shares the same TRR pattern and the positional encoding as $\tilde{\boldsymbol{p}}_{query}$, we can obtain

$$\begin{aligned}
&(\sum_{i=1}^l\text{softmax}(\tilde{\boldsymbol{p}_i}^\top\boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\boldsymbol{W}_V\tilde{\boldsymbol{p}}_i-\boldsymbol{z}^n)^\top\sum_{i=1}^l\text{softmax}(\tilde{\boldsymbol{p}_i}^\top\boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\boldsymbol{W}_V\tilde{\boldsymbol{p}}_i\\
&\cdot\tilde{\boldsymbol{p}'}^\top(\tilde{\boldsymbol{p}}_i-\sum_{r=1}^l\text{softmax}(\tilde{\boldsymbol{p}_r}^\top\boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\tilde{\boldsymbol{p}_r})\tilde{\boldsymbol{p}}_{query}^\top\tilde{\boldsymbol{p}}\\
\gtrsim&-2(1-p_n(t))^2p_n(t).
\end{aligned}\tag{175}$$

We next consider the case where $\tilde{\boldsymbol{p}}$ shares the same TRR pattern and the different positional encoding as $\tilde{\boldsymbol{p}}_{query}$. Then,

$$\begin{aligned}
&(\sum_{i=1}^l\text{softmax}(\tilde{\boldsymbol{p}_i}^\top\boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\boldsymbol{W}_V\tilde{\boldsymbol{p}}_i-\boldsymbol{z}^n)^\top\sum_{i=1}^l\text{softmax}(\tilde{\boldsymbol{p}_i}^\top\boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\boldsymbol{W}_V\tilde{\boldsymbol{p}}_i\\
&\cdot\tilde{\boldsymbol{p}'}^\top(\tilde{\boldsymbol{p}}_i-\sum_{r=1}^l\text{softmax}(\tilde{\boldsymbol{p}_r}^\top\boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\tilde{\boldsymbol{p}_r})\tilde{\boldsymbol{p}}_{query}^\top\tilde{\boldsymbol{p}}\\
\gtrsim&-(1-p_n(t))(1-p_n(t))p_n(t).
\end{aligned}\tag{176}$$

We next consider the case where $\tilde{\boldsymbol{p}}$ shares the same positional encoding and the different TRR pattern as $\tilde{\boldsymbol{p}}_{query}$. Then,

$$\begin{aligned}
&\Big|(\sum_{i=1}^l\text{softmax}(\tilde{\boldsymbol{p}_i}^\top\boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\boldsymbol{W}_V\tilde{\boldsymbol{p}}_i-\boldsymbol{z}^n)^\top\sum_{i=1}^l\text{softmax}(\tilde{\boldsymbol{p}_i}^\top\boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\boldsymbol{W}_V\tilde{\boldsymbol{p}}_i\\
&\cdot\tilde{\boldsymbol{p}'}^\top(\tilde{\boldsymbol{p}}_i-\sum_{r=1}^l\text{softmax}(\tilde{\boldsymbol{p}_r}^\top\boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\tilde{\boldsymbol{p}_r})\tilde{\boldsymbol{p}}_{query}^\top\tilde{\boldsymbol{p}}\Big|\\
\lesssim&\epsilon.
\end{aligned}\tag{177}$$

Therefore,

$$\begin{aligned}
&\Big|\tilde{\boldsymbol{p}'}^\top \eta \frac{1}{B}\sum_{n\in\mathcal{B}_b}\frac{\partial \ell(\Psi;\boldsymbol{P}^n,\boldsymbol{z}^n)}{\partial \boldsymbol{W}}\boldsymbol{p}\Big|\\
=&\Big|\eta\frac{1}{B}\sum_{n\in\mathcal{B}_b}(F(\Psi;\boldsymbol{P})-\boldsymbol{z}^n)^\top\sum_{i=1}^l \boldsymbol{W}_V\tilde{\boldsymbol{p}}_i\text{softmax}(\tilde{\boldsymbol{p}_i}^\top\boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\tilde{\boldsymbol{p}}^\top\\
&\cdot(\tilde{\boldsymbol{p}}_i-\sum_{r=1}^l\text{softmax}(\tilde{\boldsymbol{p}_r}^\top\boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\tilde{\boldsymbol{p}_r})\tilde{\boldsymbol{p}}_{query}^\top\tilde{\boldsymbol{p}}\Big|\\
\leq&\eta\epsilon.
\end{aligned}\tag{178}$$

We next discuss when $\tilde{\boldsymbol{p}}$ only shares the same positional encoding as $\tilde{\boldsymbol{p}}'$. When $l \geq \Omega(\alpha^{-1})$, and when $\tilde{\boldsymbol{p}}$ shares the same TRR pattern and the positional encoding as $\tilde{\boldsymbol{p}}_{query}$,

$$\begin{aligned}
&(\sum_{i=1}^l \text{softmax}(\tilde{\boldsymbol{p}_i}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\boldsymbol{W}_V\tilde{\boldsymbol{p}}_i - \boldsymbol{z}^n)^\top \sum_{i=1}^l \text{softmax}(\tilde{\boldsymbol{p}_i}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\boldsymbol{W}_V\tilde{\boldsymbol{p}}_i\\
&\cdot \tilde{\boldsymbol{p}'}^\top(\tilde{\boldsymbol{p}}_i - \sum_{r=1}^l \text{softmax}(\tilde{\boldsymbol{p}_r}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\tilde{\boldsymbol{p}_r})\tilde{\boldsymbol{p}}_{query}^\top\tilde{\boldsymbol{p}}\\
\lesssim & \epsilon.
\end{aligned} \tag{179}$$

We next consider the case where $\tilde{\boldsymbol{p}}$ shares the same TRR pattern and the different positional encoding as $\tilde{\boldsymbol{p}}_{query}$. Then,

$$\begin{aligned}
&(\sum_{i=1}^l \text{softmax}(\tilde{\boldsymbol{p}_i}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\boldsymbol{W}_V\tilde{\boldsymbol{p}}_i - \boldsymbol{z}^n)^\top \sum_{i=1}^l \text{softmax}(\tilde{\boldsymbol{p}_i}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\boldsymbol{W}_V\tilde{\boldsymbol{p}}_i\\
&\cdot \tilde{\boldsymbol{p}'}^\top(\tilde{\boldsymbol{p}}_i - \sum_{r=1}^l \text{softmax}(\tilde{\boldsymbol{p}_r}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\tilde{\boldsymbol{p}_r})\tilde{\boldsymbol{p}}_{query}^\top\tilde{\boldsymbol{p}}\\
\lesssim & -p_n(t)(1-p_n(t))(-1+p_n(t)) + \frac{1}{M}\\
\lesssim & p_n(t)(1-p_n(t))^2.
\end{aligned} \tag{180}$$

We next consider the case where $\tilde{\boldsymbol{p}}$ shares the same positional encoding and the different TRR pattern as $\tilde{\boldsymbol{p}}_{query}$. Then,

$$\begin{aligned}
&(\sum_{i=1}^l \text{softmax}(\tilde{\boldsymbol{p}_i}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\boldsymbol{W}_V\tilde{\boldsymbol{p}}_i - \boldsymbol{z}^n)^\top \sum_{i=1}^l \text{softmax}(\tilde{\boldsymbol{p}_i}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\boldsymbol{W}_V\tilde{\boldsymbol{p}}_i\\
&\cdot \tilde{\boldsymbol{p}'}^\top(\tilde{\boldsymbol{p}}_i - \sum_{r=1}^l \text{softmax}(\tilde{\boldsymbol{p}_r}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\tilde{\boldsymbol{p}_r})\tilde{\boldsymbol{p}}_{query}^\top\tilde{\boldsymbol{p}}\\
\lesssim & \epsilon.
\end{aligned} \tag{181}$$

Therefore,

$$\begin{aligned}
&\tilde{\boldsymbol{p}'}^\top \eta\frac{1}{B}\sum_{n\in\mathcal{B}_b}\frac{\partial \ell(\Psi;\boldsymbol{P}^n,\boldsymbol{z}^n)}{\partial \boldsymbol{W}}\boldsymbol{p}\\
=&\eta\frac{1}{B}\sum_{n\in\mathcal{B}_b}(F(\Psi;\boldsymbol{P})-\boldsymbol{z}^n)^\top\sum_{i=1}^l \boldsymbol{W}_V\tilde{\boldsymbol{p}}_i\text{softmax}(\tilde{\boldsymbol{p}_i}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\tilde{\boldsymbol{p}}^\top\\
&\cdot(\tilde{\boldsymbol{p}}_i - \sum_{r=1}^l \text{softmax}(\tilde{\boldsymbol{p}_r}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\tilde{\boldsymbol{p}_r})\tilde{\boldsymbol{p}}_{query}^\top\tilde{\boldsymbol{p}}\\
\gtrsim&\eta\frac{1}{B}\sum_{n\in\mathcal{B}_b}\frac{1}{2M}\cdot p_n(b)(1-p_n(b))^2.
\end{aligned} \tag{182}$$

We then consider if $\tilde{\boldsymbol{p}}$ shares a different TRR pattern and a different positional encoding as $\tilde{\boldsymbol{p}}'$. When $l \geq \Omega(\alpha^{-1})$, and when $\tilde{\boldsymbol{p}}$ shares the same TRR pattern and the positional encoding as $\tilde{\boldsymbol{p}}_{query}$,

$$\begin{aligned}
&(\sum_{i=1}^l \text{softmax}(\tilde{\boldsymbol{p}_i}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\boldsymbol{W}_V\tilde{\boldsymbol{p}}_i - \boldsymbol{z}^n)^\top \sum_{i=1}^l \text{softmax}(\tilde{\boldsymbol{p}_i}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\boldsymbol{W}_V\tilde{\boldsymbol{p}}_i\\
&\cdot \tilde{\boldsymbol{p}'}^\top(\tilde{\boldsymbol{p}}_i - \sum_{r=1}^l \text{softmax}(\tilde{\boldsymbol{p}_r}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\tilde{\boldsymbol{p}_r})\tilde{\boldsymbol{p}}_{query}^\top\tilde{\boldsymbol{p}}\\
\gtrsim & \epsilon.
\end{aligned} \tag{183}$$

We next consider the case where $\tilde{\boldsymbol{p}}$ shares the same TRR pattern and the different positional encoding as $\tilde{\boldsymbol{p}}_{query}$. Then,

$$\begin{aligned}
&(\sum_{i=1}^l \text{softmax}({\tilde{\boldsymbol{p}}_i}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\boldsymbol{W}_V\tilde{\boldsymbol{p}}_i-\boldsymbol{z}^n)^\top\sum_{i=1}^l \text{softmax}({\tilde{\boldsymbol{p}}_i}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\boldsymbol{W}_V\tilde{\boldsymbol{p}}_i\\
&\cdot\tilde{\boldsymbol{p}'}^\top(\tilde{\boldsymbol{p}}_i-\sum_{r=1}^l \text{softmax}({\tilde{\boldsymbol{p}}_r}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\tilde{\boldsymbol{p}}_r)\tilde{\boldsymbol{p}}_{query}^\top\tilde{\boldsymbol{p}}\\
\gtrsim&-(1-p_n(t))p_n(t).
\end{aligned}\tag{184}$$

We next consider the case where $\tilde{\boldsymbol{p}}$ shares the same positional encoding and the different TRR pattern as $\tilde{\boldsymbol{p}}_{query}$. Then,

$$\begin{aligned}
&\Big|(\sum_{i=1}^l \text{softmax}({\tilde{\boldsymbol{p}}_i}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\boldsymbol{W}_V\tilde{\boldsymbol{p}}_i-\boldsymbol{z}^n)^\top\sum_{i=1}^l \text{softmax}({\tilde{\boldsymbol{p}}_i}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\boldsymbol{W}_V\tilde{\boldsymbol{p}}_i\\
&\cdot\tilde{\boldsymbol{p}'}^\top(\tilde{\boldsymbol{p}}_i-\sum_{r=1}^l \text{softmax}({\tilde{\boldsymbol{p}}_r}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\tilde{\boldsymbol{p}}_r)\tilde{\boldsymbol{p}}_{query}^\top\tilde{\boldsymbol{p}}\Big|\\
\lesssim&\epsilon.
\end{aligned}\tag{185}$$

Therefore,

$$\begin{aligned}
&\Big|\tilde{\boldsymbol{p}'}^\top\eta\frac{1}{B}\sum_{n\in\mathcal{B}_b}\frac{\partial\ell(\Psi;\boldsymbol{P}^n,\boldsymbol{z}^n)}{\partial\boldsymbol{W}}\boldsymbol{p}\Big|\\
=&\Big|\eta\frac{1}{B}\sum_{n\in\mathcal{B}_b}(F(\Psi;\boldsymbol{P})-\boldsymbol{z}^n)^\top\sum_{i=1}^l\boldsymbol{W}_V\tilde{\boldsymbol{p}}_i\text{softmax}({\tilde{\boldsymbol{p}}_i}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\tilde{\boldsymbol{p}}^\top\\
&(\tilde{\boldsymbol{p}}_i-\sum_{r=1}^l \text{softmax}({\tilde{\boldsymbol{p}}_r}^\top \boldsymbol{W}\tilde{\boldsymbol{p}}_{query})\boldsymbol{p}_r)\tilde{\boldsymbol{p}}_{query}^\top\tilde{\boldsymbol{p}}\Big|\\
\lesssim&\eta\epsilon.
\end{aligned}\tag{186}$$

$\square$