ON THE CAPABILITY AND LIMITATION OF HARD PROMPT

Anonymous authors
Paper under double-blind review

ABSTRACT

Prompt engineering has become an indispensable tool for using large language models (LLMs), turning LLMs into task-specific experts without changing their weights. Despite notable theoretical advances in prompt engineering, the theory for the more practical hard or discrete prompts is largely open. In this paper, we try to fill this gap either by providing a complete solution or by making substantial progress on the three core theoretical questions regarding hard prompts. First, we show that determining the existence of a hard prompt for a transformer to solve a downstream task is NP-complete and that finding an optimal hard prompt is NP-hard, which is the first computational complexity result for hard prompting, as far as we know. Second, we show that, unlike soft or continuous prompts, hard prompts have essential limitations: hard prompts are not complete; short hard prompts do not significantly enhance the ability of transformers; and long hard prompts exhibit the "prompt dominating answer phenomenon," meaning that, with high probability, the same answer is given for all queries of the same length. On the other hand, linear hard prompts do not have the limitations of short or long prompts. Third, we provide a tight bound on the size of the task in terms of the prompt length for the performance of prompts on the finite task to generalize to the entire data distribution, leading to a necessary and sufficient condition for generalizability. This is the first result on generalization for prompting, as far as we know. Our findings not only offer the first theoretical insights into hard prompts but also provide provably reliable practical guidance for real-world LLM usage.

1 INTRODUCTION

Prompt engineering has become a pivotal technique for the great success of large language models (LLMs), enhancing their accuracy, reasoning ability, and applicability across almost all areas (Brown et al., 2020; Wei et al., 2022; Shin et al., 2020; Gao et al., 2021; Debnath et al., 2025). Prompts for LLMs can be broadly categorized into two paradigms: hard (discrete) prompts and soft (continuous) prompts (Li et al., 2025). Hard prompts refer to discrete instructions composed of real natural language tokens, which are human-readable and require no weight changes (Li et al., 2025). In contrast, soft prompts adopt learnable continuous embedding vectors inserted into the model input, which freeze the backbone model and only optimize a small number of parameters (Li & Liang, 2021; Lester et al., 2021; Li et al., 2024). Hard prompts are the earliest and most straightforward practical form of prompt design, while soft prompts have become a prevalent, lightweight tuning strategy for large language models. In practice, prompts can either be designed by human experts (Brown et al., 2020; Wei et al., 2022) or found through prompt optimization (Li & Liang, 2021; Lester et al., 2021; Prasad et al., 2022; Zhang et al., 2022; Pryzant et al., 2023) by minimizing the loss.

Besides the practical progress, substantial efforts have been directed towards establishing a theoretical framework for prompt engineering. Prompts have been proven to be complete in a certain sense. By modifying the embedding layer, soft prompts can turn a transformer model into a perfect memorizer of a finite dataset (Wang & Lin, 2025; Hu et al., 2025). Using a suitably constructed transformer, prompts are Turing complete for emulating computable functions (Qiu et al., 2025). Specific transformers can be designed such that, when equipped with prompts, they can serve as universal approximators for certain functions (Wang et al., 2023; Petrov et al., 2024; Nakada et al., 2025; Hu et al., 2025). Theoretical results for prompts in the form of in-context learning were given in (Shen et al., 2024; Yang et al., 2024; Jeon et al., 2024; Hendel et al., 2023; Xie et al., 2021).

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

However, the theoretical understanding of hard prompts remains largely underexplored. In this paper, we investigate three core theoretical questions regarding hard prompts and provide theoretical explanations for several questions and phenomena encountered in prompt engineering.

Question 1: What is the computational complexity of computing an optimal hard prompt for a given transformer and task?

Let $\mathcal{F}$ be a transformer, $S = \{(x_i, y_i)\}_{i=1}^N$ a downstream task consisting of query-answer pairs, and $\mathcal{L}(\mathcal{F}, S)$ the supervised loss of $\mathcal{F}$ on S. A sequence $\mathcal{P}$ is called a *prompt of $\mathcal{F}$ for S with loss α* if $\mathcal{L}(\mathcal{F}, \mathcal{P} \oplus S) \leq \alpha$, where $\mathcal{P} \oplus S = \{(\mathcal{P} \oplus x_i, y_i)\}_{i=1}^N$ and $\mathcal{P} \oplus x_i$ are the concatenation of $\mathcal{P}$ and x_i.

Further, $\mathcal{P}$ is called a *polynomial prompt* if the length of $\mathcal{P}$ is polynomial in that of S. We thus have (see Theorem 4.2 and Corollary 4.5):

Theorem 1.1 (Informal)**.** *Determining the existence of a polynomial hard prompt $\mathcal{P}$ of $\mathcal{F}$ for S with a given loss is NPC, and finding a polynomial hard prompt with a minimum loss is NP-hard.*

Since $\mathcal{L}(\mathcal{F}, \mathcal{P} \oplus S)$ is the optimization objective in prompt optimization, this result indicates that finding an optimal polynomial hard prompt is an NP-hard problem. To the best of our knowledge, this is the first theoretical result on the computational complexity of hard prompting.

It is reasonable to consider polynomial-length prompts, because super-polynomial prompts are computationally intractable in practice and are already NP-hard to compute in terms of the task.

Question 2: What are the powers and limitations of hard prompts in enhancing the inference ability of transformers?

Soft prompts have been demonstrated to be complete in the memorization of a finite dataset (Hu et al., 2025). In contrast, we demonstrate that hard prompts face fundamental limitations.

First, we show that hard prompts are, in a certain sense, incomplete (see Theorem 5.2).

Theorem 1.2 (Informal)**.** *Polynomial hard prompts are not complete in that they cannot help a transformer with fixed weights to solve all tasks with a given loss. Specifically, for certain query set $S_q = \{x_i\}_{i=1}^N$ and $\alpha > 0$, there exist no transformers $\mathcal{F}$ that can solve task $S = \{(x_i, y_i)\}_{i=1}^N$ for any answer set $S_a = \{y_i\}_{i=1}^N$ by using a polynomial prompt $\mathcal{P}$.*

This theoretical finding highlights fundamental distinctions between soft prompts and hard prompts. Note that this is not contradictory to previous work (Hu et al., 2025; Qiu et al., 2025; Wang & Lin, 2025). Refer to Remark 5.3 for a detailed comparison. We further have

Theorem 1.3 (Informal)**.** *Let the length of the prompt $\mathcal{P}$ be polynomial in that of the task S.*

(1) When the prompts are much shorter than the task, they do not significantly enhance a transformer's performance on that task. More specifically, if a transformer is unable to solve a long task without any prompt, then, with high probability, it will also fail to solve that task when given only a short prompt (see Theorem 5.4 and Corollary 5.5).

(2) When the prompts are much longer than the task, they display the "Prompt Dominating Answer Phenomenon", which means that the transformer produces an identical answer—taken directly from the prompt itself—for every query with the same length and last token (see Theorem 5.7).

(3) Linear prompts avoid the previously mentioned drawbacks of both short and long prompts, and they can boost the effectiveness of certain transformers on certain tasks with high probability (see Propositions 5.11 and 5.13).

This result provides an explanation for the possible problems that may occur in practice when using prompts of different lengths. By (1), a short prompt cannot solve long tasks with high probability because the prompt has a limited effect on the later part of the output. By (2), a long prompt can cause the transformer to ignore the query itself with a high probability. This shows a key distinction between prompting and training, where the scaling law (Kaplan et al., 2020) implies that more data and parameters lead to better performance, but a long prompt does not.

Question 3: Can the effects of prompts for a finite task be generalized to additional query-answer samples?

We provide a complete answer to this problem (Refer to Theorems 6.1 and 6.3).

Theorem 1.4 (Informal)**.** *Let S be a finite task drawn iid from a distribution $\mathcal{D}$. We provide upper and lower bounds of $|S|$ in terms of $\mathrm{len}(\mathcal{P})$ for the performance of $\mathcal{P}$ on S to generalize to $\mathcal{D}$. Both bounds are $\overline{O}(\mathrm{len}(\mathcal{P}))$ when omitting some small quantities, resulting in a tight generalization bound.*

This result indicates that a necessary and sufficient condition for generalizability is that the size of the task should be larger than the length of the prompt.

Summary. We either provide answers or make substantial contributions to the three aforementioned questions. The main contributions of this paper are:

- We provide a complete solution to Question 1. Determining the existence of polynomial hard prompts for a finite task is NP-complete, and obtaining optimal ones is NP-hard.
- We make substantial progress on Question 2 by showing that polynomial prompts have essential limitations, as demonstrated in Theorems 1.2 and 1.3.
- We provide a complete solution to Question 3 by presenting a tight generalization bound.

Our results not only provide the first theoretical insights on hard prompts but also have practical implications: keep prompt length proportional to answer length and avoid extremely short or long prompts; a necessary sample size is required to ensure the generalization of prompts. Refer to Appendix H for more details. This paper is primarily theoretical. In Appendix J, we include simple experimental results that support our key theoretical results, serving as minimal proof-of-concept demonstrations rather than comprehensive large-scale benchmarks.

2 Related work

Prompt engineering. Prompt engineering refers to methods that guide LLMs to produce desired output by designing prompts without modifying the model (Debnath et al., 2025). Radford et al. (2019) demonstrated that zero-shot prompting can elicit task behavior. Brown et al. (2020) showed that LLMs can tackle new tasks via few-shot prompting without fine-tuning. Schick & Schütze (2020) showed that prompting also works on small LLMs. Lester et al. (2021) showed that a simple prompt is comparable to full parameter tuning. Wei et al. (2022) presented CoT prompting to enhance the reasoning capabilities of LLMs, leading to numerous powerful methods such as tree of thoughts (Yao et al., 2023) and graph of thoughts (Besta et al., 2024). RAG used external sources as prompts to increase accuracy (Lewis et al., 2020). Prompt optimization is used to generate hard prompts. Li & Liang (2021); Pryzant et al. (2023) showed that prompts can be found using gradients. Prasad et al. (2022) provided gradient-free local searches. Zhang et al. (2022) used reinforcement learning to create prompts. Sun et al. (2022) employed a black-box method to find prompts. Soft prompts, first formalized and explored in a series of seminal studies Qin & Eisner (2021); Li & Liang (2021); Liu et al. (2021); Lester et al. (2021), are learnable continuous embedding vectors inserted into transformer inputs instead of discrete text prompts. Wen et al. (2023) employed the gradient method for the embedding layer to obtain soft prompts.

Our theoretical results differ from prompt optimization, which is a practical technique for obtaining improved prompts by minimizing the loss. On the other hand, we provide rigorously proven results: hard prompt optimization is NP-hard, hard prompts are not complete, both short and long herd prompts have limitations, and a tight bound on prompt generalization is given. Our findings can be used to guide prompt optimization by helping to properly choose the prompt length.

Theory of prompting. Theory of prompting was studied from several aspects. Wang et al. (2023); Petrov et al. (2024); Hu et al. (2025); Nakada et al. (2025) demonstrated the expressive ability of transformers with prompts to approximate Lipschitz functions or certain differentiable functions. The theoretical results of prompts on ICL were presented: Shen et al. (2024) provided a convergence proof for a single head, single-layer softmax transformer; Yang et al. (2024) explained ICL from the perspective of representation alignment; Hendel et al. (2023) formulated ICL as a learning problem within a hypothesis space; Jeon et al. (2024); Xie et al. (2021) used information theory to provide bounds for the Bayesian error. Bhargava et al. (2023) studied whether prompts with $\leq k$ tokens push the output distribution closer to the target distribution using control theory. Wang et al. (2023); Meyer et al. (2025); Hu et al. (2025) studied the limitations of prompts in single-head transformers with a single self-attention layer. Hu et al. (2025); Wang & Lin (2025); Qiu et al. (2025) proved that prompts are complete for transformers to simulate a function or memorize a dataset.

3 PREREQUISITE

We will introduce the basic symbols in this section. The details are in the Appendix A.

3.1 DATA AND AUTOREGRESSIVE TRANSFORMER

Data. Let $\Sigma = \{\sigma_i\}_{i=0}^{T}$ be a set of vocabulary or tokens and Σ^* the set of all sequences of the form $(\sigma_{k_j})_{j=1}^{n}$. We assume $T \geq 3$ in this paper, which is reasonable. The symbol σ_0 is used only to stop the output of the transformer and does not appear in the middle of a sequence. For a sequence x, let $\mathrm{len}(x)$ be its length, $x[i]$ be the i-th symbol in x, and $\mathrm{last}(x) = x[\mathrm{len}(x)]$ be the last symbol of x. For $x_1, x_2 \in \Sigma^*$, we use $x_1 \oplus x_2$ to denote the concatenation of x_1 and x_2. For a finite subset S of Σ^*, denote $\mathrm{len}(S) = \max_{x \in S} \mathrm{len}(x)$.

An autoregressive transformer $\mathcal{F}$ has three parts: the embedding, hidden layers, and the output layer.

Embedding. The transformer first embeds each token σ_i into a vector $v_i \in \mathbb{R}^w$, where v_i serves as an adjustable parameter and w is called the *embedding dimension*. Then a sequence $x = (\sigma_{k_j})_{j=1}^{n} \subset \Sigma^*$ is embedded into a matrix in $\mathbb{R}^{n \times w}$: $(v_{k_1}, v_{k_2}, \ldots, v_{k_n})^\tau$, which is the input to the first hidden layer. We will use x and its matrix representation interchangeably whenever this does not lead to confusion.

Hidden layer. We first define the feedforward layer. For $x \in \mathbb{R}^{n \times w}$ of length n, the feedforward layer with width W is $\mathrm{FNN}(x) = \mathrm{Relu}(xE_1 \oplus b)E_2$, where $E_1 \in \mathbb{R}^{w \times W}, b \in \mathbb{R}^{1 \times W}, E_2 \in \mathbb{R}^{W \times w}$ are the parameters, and $xE_1 \oplus b$ means adding b to each row of xE_1. For an input $x \in \mathbb{R}^{n \times w}$, the attention-layer with width W and head H is $\mathrm{ATT}(x) = \sum_{i=1}^{H} \mathrm{softmax}(R(xQ_i, K_i x^\tau) + M)xV_i$, where $Q_i \in \mathbb{R}^{w \times W}, K_i \in \mathbb{R}^{W \times w}, V_i \in \mathbb{R}^{w \times w}$ are parameters. $M \in \{-\infty, 0\}^{n \times n}$ is a causal mask defined as $M_{i,j} = -\infty$ if and only if $j > i$. $R(\cdot, \cdot)$ means relative position embedding. Without loss of generality, we take $W = w$.

Let $x^{(0)}$ be the input. Then the output of the i-th hidden layer of the transformer is

$$x^{(i)} = x^{(i-1)} + \mathrm{ATT}_i(x^{(i-1)}) + \mathrm{FNN}_i(x^{(i-1)} + \mathrm{ATT}_i(x^{(i-1)}))$$

where ATT_i and FNN_i are the i-th attention and feedforward layers.

Output layer. The output layer performs a linear transformation on the last row of the output from the last hidden layer, written as $x^{(L)}[n]$, that is: $\mathcal{F}(x) = x^{(L)}[n]H + c \in \mathbb{R}^{T+1}$, where $H \in \mathbb{R}^{w \times (T+1)}$ and $c \in \mathbb{R}^{1 \times (T+1)}$ are the parameters of the output layer. Let $\pi_{\mathcal{F}}(\cdot \mid x) = \mathrm{softmax}(\mathcal{F}(x))$, which can be saw as a distribution over Σ.

Output form. The final output $y = \widehat{\mathcal{F}}(x)$ is obtained as follows, with the initial value $y = ()$: (1) The next token $s \in \Sigma$ be randomly sampled according to the distribution $\pi_{\mathcal{F}}(\cdot \mid x \oplus y)$; (2) If $s \neq \sigma_0$, then let $y = y \oplus \{s\}$ and return to step (1); (3) If $s = \sigma_0$, stop and return the output $y = \widehat{\mathcal{F}}(x)$.

Given the input x, the probability that $\mathcal{F}$ outputs y is

$$\pi_{\mathcal{F}}(y \mid x) = \prod_{i=1}^{\mathrm{len}(y)+1} \pi_{\mathcal{F}}(y[i] \mid x \oplus y[0:i-1]) \tag{1}$$

where $y[0:0] = ()$ and $\mathrm{last}(y) = \sigma_0$. Let $\mathcal{L}_{\mathrm{nll}}(\mathcal{F}, x, y) = -\log \pi_{\mathcal{F}}(y \mid x)$ represent the negative log-likelihood (Bengio et al., 2000). Note that stochastic sampling strategies, such as temperature sampling (Ficler & Goldberg, 2017), have been widely used.

3.2 POLYNOMIAL-LENGTH PROMPT

In this paper, a *task* is a finite set of query-answer pairs: $S = \{(x_i, y_i)\}_{i=1}^{N} \subset \Sigma^* \times \Sigma^*$, where $S_q = \{x_i\}_{i=1}^{N}$ is the *query set* and $S_a = \{y_i\}_{i=1}^{N}$ is the *answer set*. For a prompt $\mathcal{P} \in \Sigma^*$, denote $\mathcal{P} \oplus S = \{(\mathcal{P} \oplus x_i, y_i)\}_{i=1}^{N}$ and $\mathcal{P} \oplus S_q = \{\mathcal{P} \oplus x_i\}_{i=1}^{N}$.

Remark 3.1. In this paper, a single prompt $\mathcal{P}$ is used for all query-answer pairs of a task S, which is a standard practice (Wang & Lin, 2025; Qiu et al., 2025). For instance, Goedel-Prover (Lin et al., 2025) uses a single prompt, while Azerbayev et al. (2024) uses a single prompt for each benchmark set.

Prompt with a given loss. Following the common practice in machine learning, we will use the value of the training loss as a measure of the prompt's success. The commonly used crossentropy loss for $\mathcal{F}$ on a task $S = \{(x_i, y_i)\}_{i=1}^N$ is $\mathcal{L}(\mathcal{F}, S) = \frac{1}{N}\sum_{i=1}^N \mathcal{L}_{\text{nll}}(\mathcal{F}, x_i, y_i)$.

$\mathcal{P} \in \Sigma^*$ is called *a prompt of transformer $\mathcal{F}$ for a task S with loss $\alpha \in \mathbb{R}_{>0}$* if $\mathcal{L}(\mathcal{F}, \mathcal{P} \oplus S) \leq \alpha$. For convenience, we also say that $\mathcal{F}$ *"solves" task S using prompt $\mathcal{P}$ with loss α*.

Remark 3.2. In machine learning, the loss widely serves as a surrogate for accuracy in training and in generalization bounds (Mohri et al., 2018), because a lower loss value generally indicates higher accuracy. For example, for the single data case, the probability of $y = \widehat{\mathcal{F}}(x)$ is $\pi_{\mathcal{F}}(y \mid x) = \exp^{-\mathcal{L}_{\text{nll}}(\mathcal{F}, x, y)}$ which is higher when the loss is smaller. Refer to Appendix I for more details.

Polynomial-length prompt with loss. To be computationally tractable, we consider prompts of polynomial size, leading to the following main concept of this paper. We say that a transformer $\mathcal{F}$ *"solves" task S with loss α using a (c, d)-prompt* $\mathcal{P} \in \Sigma^*$ if $\mathcal{L}(\mathcal{F}, \mathcal{P} \oplus S) \leq \alpha$ and $\text{len}(\mathcal{P}) \leq c\,\text{len}(S_a)^d$, where $\alpha, c, d \in \mathbb{R}_{>0}$. $\mathcal{P}$ is also called *a (α, c, d)-prompt* or *a polynomial prompt*.

Remark 3.3. We focus on polynomial-length prompts because super-polynomial prompts are computationally intractable in practice.

Remark 3.4. Hard prompts are discrete natural language instructions that do not involve adapting model weights. Therefore, a fixed transformer is used in our problem formulation.

4 Computational complexity for hard prompting

In this section, we show that finding a polynomial-length hard prompt with a given loss is an NP-hard problem. Some supplementary instructions are in Appendix B.1. Let $\mathcal{F}$ be a transformer and $S = \{(x_i, y_i)\}_{i=1}^N \subset \Sigma^* \times \Sigma^*$ a task with N elements. We define the following problem:

Definition 4.1. The decision problem $\mathbf{Prompt}(\mathcal{F}, S, \alpha, c, d)$ is: for a transformer $\mathcal{F}$, a task S, and $\alpha, c, d \in \mathbb{R}_{>0}$, determine whether there exists an (α, c, d)-prompt $\mathcal{P} \in \Sigma^*$ of $\mathcal{F}$ for S?

Then we have

Theorem 4.2. $\mathbf{Prompt}(\mathcal{F}, S, \alpha, c, d)$ *is an NPC problem. Thus, finding an (α, c, d)-prompt of $\mathcal{F}$ for S is NP-hard.*

Proof Idea. Given a 3-SAT problem with N variables and M Bohr expressions, we can construct a symbol set Σ, $\alpha, c, d \in \mathbb{R}_{>0}$, a transformer $\mathcal{F}$, and a task S with size $\text{poly}(N, M)$ such that the existence of an (α, c, d)-prompt of $\mathcal{F}$ for S is computationally equivalent to this 3-SAT problem.

Based on the proof of Theorem 4.2, we can deduce the following result.

Corollary 4.3. $\mathbf{Prompt}(\mathcal{F}, S, \alpha, c, d)$ *is NPC for the single pair task $S = \{(x_1, y_1)\}$.*

In practice, effective hard prompts can be obtained with optimization (Shin et al., 2020; Sabbatella et al., 2024). We formulate our problem as the following optimization problem.

Definition 4.4. For a transformer $\mathcal{F}$, task S, and $c, d \in \mathbb{R}_{>0}$, the problem $\text{DPO}(\mathcal{F}, S, c, d)$ is to solve the following optimization problem to obtain a prompt $\mathcal{P}$:

$$\text{argmin}_{\mathcal{P} \in \Sigma^*, \text{len}(\mathcal{P}) \leq c\,\text{len}(S_a)^d}\ \mathcal{L}(\mathcal{F}, \mathcal{P} \oplus S).$$

As a direct consequence of Theorem 4.2, we have

Corollary 4.5. $\text{DPO}(\mathcal{F}, S, c, d)$ *is an NP-hard problem.*

When the greedy sampling strategy is applied to next token prediction—meaning the next token σ is chosen as the one that maximizes $\pi_{\mathcal{F}}(\sigma \mid x \oplus y)$—the transformer produces a deterministic output, denoted by $y = \overline{\mathcal{F}}(x)$. We show that finding prompts in this case is also NPC. Refer to Appendix B.3 for a more detailed discussion.

Proposition 4.6. *For any transformer $\mathcal{F}$, task $S = \{(x_i, y_i)\}_{i=1}^N$, and $c, d \in \mathbb{R}_{>0}$, deciding whether there exists a prompt $\mathcal{P} \in \Sigma^*$ such that $y_i = \overline{\mathcal{F}}(\mathcal{P} \oplus x_i)$ for $i = 1, \ldots, N$ and $\text{len}(\mathcal{P}) \leq c\,\text{len}(S_a)^d$ is NPC. The result is also true for $N = 1$.*

5 EXPRESSIVE LIMITATION AND CAPABILITIES OF HARD PROMPT

Prompts can greatly enhance the expressive power of transformers. However, empirical studies show that prompts also have certain limitations (Wang et al., 2023; Meyer et al., 2025). In this section, we will provide a theoretical analysis of the capabilities and limitations of hard prompts. Practical implications of our theoretical results in this section are provided in Appendix H.

5.1 POLYNOMIAL HARD PROMPTS ARE NOT COMPLETE

In this section, we show that polynomial hard prompts are not complete in that they cannot help a fixed transformer solving all tasks with a given loss.

We begin with a positive result, showing that transformers equipped with prompts can solve most tasks of bounded length, provided that the transformers are designed with knowledge of the query set.

Proposition 5.1. *For any query set $S_q = \{x_i\}_{i=1}^N \subset \Sigma^*$ satisfying $x_i \neq x_j[1 : \text{len}(x_i)]$ for all $i \neq j \in [N]$, and $\alpha > 0, L \geq 2, c \geq N \log_T(T+1) + \log_T L + 2, d > 1$, there exists a transformer $\mathcal{F}$ that can solve task $S = \{(x_i, y_i)\}_{i\in[N]}$ with an (α, c, d)-prompt for any answer set $S_a = \{y_i\}_{i=1}^N \subset \Sigma^*$ satisfying $\text{len}(S_a) \leq L$.*

The condition $x_i \neq x_j[1 : \text{len}(x_i)]$ is easily achievable, by adding a punctuation mark. More interestingly, if there is no length limitation for the answer set, for some query sets that satisfy the condition in Proposition 5.1, it is impossible to use a prompt to solve all answers with a given loss.

Theorem 5.2. *For some query set $S_q = \{x_i\}_{i=1}^N \subset \Sigma^*$ satisfying $x_i \neq x_j[1 : \text{len}(x_i)]$ for all $i, j \in [N]$ and $\alpha > 0$, there exist no transformer $\mathcal{F}$ and $c > 0, d > 1$ such that $\mathcal{F}$ can solve task $S = \{(x_i, y_i)\}_{i=1}^N$ with an (α, c, d)-prompt for any answer set $S_a = \{y_i\}_{i=1}^N \subset \Sigma^*$.*

Proof Idea: We can construct two queries x, x' such that for any transformer $\mathcal{F}$, the output sequences of $\mathcal{F}(\mathcal{P} \oplus x)$ and $\mathcal{F}(\mathcal{P} \oplus x')$ will gradually become similar as the length of the prompt $\mathcal{P}$ increases, resulting in $\mathcal{F}$ being unable to achieve a small loss in both samples simultaneously.

As a direct consequence, we can demonstrate that polynomial-length hard prompts do not enable transformers to solve all tasks with arbitrarily small loss when given the query set S_q.

Remark 5.3. Our negative results in Theorem 5.2 on hard prompts do not contradict the completeness of prompting (Wang & Lin, 2025; Qiu et al., 2025; Hu et al., 2025; Nakada et al., 2025) because the settings are quite different. More precisely, soft prompts are used in (Wang & Lin, 2025; Hu et al., 2025), which allow for changing the embedding layer; the transformer in (Qiu et al., 2025) is constructed as a certain Turing complete transformer similar to (Pérez et al., 2021); and the transformer in (Nakada et al., 2025) is also a constructed one with 7-layers.

5.2 SHORT HARD PROMPT DOES NOT HELP SOLVING LONG TASK

In this section, we will show that when the length of the prompt is shorter than that of the answer, that is $d \in (0, 1)$, the prompt has limited ability, as shown below.

Theorem 5.4. *For any transformer $\mathcal{F}$, $c > 0$, and $\epsilon, d \in (0, 1)$, there exists an $L \in \mathbb{Z}_{>0}$ such that if a task $S = \{(x_i, y_i)\}_{i=1}^N$ satisfies $\text{len}(y_i) \geq L$ for all $i \in [N]$ and $\mathcal{F}$ cannot solve S with loss $(\text{len}(S) + 1)\alpha$ without a prompt, then $\mathcal{F}$ also cannot solve S using a $(\max\{(\text{len}(S) + 1)(\alpha - \epsilon), 0\}, c, d)$-prompt.*

Proof Idea. We will show that $||\mathcal{F}(\mathcal{P} \oplus x \oplus y[1 : i]) - \mathcal{F}(x \oplus y[1 : i])||_2 \propto \frac{\text{len}(\mathcal{P})}{i^d}$ and then use it to estimate the difference between loss functions for using a prompt and without a prompt.

By Theorem 5.4, if a transformer is hard to solve a task, i.e. $\alpha \gg 0$, then it cannot solve such a task using short prompts with a smaller loss. Intuitively, this suggests that **a short prompt does little to enable the transformer to solve long tasks that it is otherwise unable to solve without a prompt.**

To better understand Theorem 5.4, we consider the single data situation and use the length normalized loss function $\mathcal{L}_{\text{Nnll}}(\mathcal{F}, x, y) = \mathcal{L}_{\text{nll}}(\mathcal{F}, x, y)/(\text{len}(y) + 1)$ (Yang et al., 2020). We have the following corollary

Corollary 5.5. *For any transformer $\mathcal{F}$ and $\epsilon, c> 0, d \in (0,1)$, there exists an $L \in \mathbb{Z}_{>0}$ such that if a task $S = \{(x,y)\}$ satisfies $\text{len}(y) \geq L$, then $|\mathcal{L}_{Nnll}(\mathcal{F}, \mathcal{P} \oplus x, y) - \mathcal{L}_{Nnll}(\mathcal{F}, x, y)| \leq \epsilon$ for any (c,d)-prompt $\mathcal{P}$.*

This corollary illustrates the reason behind Theorem 5.4: for any query x, when the answer is much longer than the prompt, the average loss will not change significantly by using a short prompt, which implies that even if the prompt has a strong impact on the early parts of the output, the effectiveness of the prompt decreases in the later portions of the output.

Since Theorem 5.4 only stands for long answers, it is natural to ask: is there an $\mathcal{F}$ that can solve any task with a fixed length using short prompts? The answer is no, as demonstrated below.

Proposition 5.6. *For any token set Σ of size T, transformer $\mathcal{F}$, query $x \in \Sigma^*$, and $\alpha > 0, c > 0, d \in (0,1), L \in \mathbb{Z}_+$ satisfying $L > (\log_T(T+1)c+\alpha)^{\frac{1}{1-d}}$, there exists a $y \in \Sigma^*$ with $\text{len}(y) = L$ such that $\mathcal{F}$ cannot solve $\{(x,y)\}$ using an (α, c, d)-prompt.*

This shows that even if the length of the answer does not depend on $\mathcal{F}$, it is still impossible to solve some tasks using a short prompt with a small loss when L is longer than a threshold independent of $\mathcal{F}$. This is because the number of selectable short prompts is fewer than the number of answers, which naturally leads to situations where solving arbitrary answers with short prompts is not possible.

5.3 Long prompt has "Prompt Dominating Answer Phenomenon"

For long prompts, that is $d > 1$, the limitations of short prompts, such as those in Theorem 5.4, no longer exist because long prompts significantly enhance their impact on the final result. However, in this section, we show that long prompts still have certain limitations: if the prompt is too long, then the output of the transformer no longer focuses on the task itself but is primarily derived from the prompt, or equivalently, the prompt "dominates" the answer. We refer to this as the "Prompt Dominating Answer Phenomenon." We consider the single data situation.

Theorem 5.7. *For any transformer $\mathcal{F}$, $\epsilon > 0$, there exists $c > 0, d > 1$ such that for any x, x', y satisfying $\text{len}(x) = \text{len}(x')$, $\text{last}(x) = \text{last}(x')$, and $\text{len}(y) > \text{len}(x)$, if $\mathcal{F}$ can solve $\{(x,y)\}$ with a loss α using a prompt $\mathcal{P}$ with a length greater than $c\text{len}(y)^d$, then $\mathcal{F}$ can also solve $\{(x',y)\}$ with a loss $\alpha + \epsilon$ by using the same prompt $\mathcal{P}$.*

Proof Idea We will show $||\mathcal{F}(\mathcal{P} \oplus x \oplus y[1:i]) - \mathcal{F}(\mathcal{P} \oplus x' \oplus y[1:i])||_2 \propto \frac{\text{len}(x)+i}{\text{len}(\mathcal{P})}$ at first. Hence, we can construct c, d such that $|\mathcal{L}_{\text{nll}}(\mathcal{F}, \mathcal{P} \oplus x, y) - \mathcal{L}_{\text{nll}}(\mathcal{F}, \mathcal{P} \oplus x', y)| \leq \epsilon$ for any prompt satisfying $\text{len}(\mathcal{P}) \geq c(\text{len}(x) + \text{len}(y))^d$, which leads to the theorem directly.

The theorem shows that when the prompt is much longer than the answer, it yields the same answer with high probability for all queries with the same length and last token. Intuitively, this result is reasonable because when the prompt is too long, the information from the query is obscured, causing the transformer to focus too much on the prompt and ignore the information in the query.

If the answer is much longer than the query, then we can provide a stronger result for any $d > 2$.

Proposition 5.8. *For any transformer $\mathcal{F}$, $\epsilon, c > 0, d > 2$, there exists an L such that for any x, x', y satisfying $\text{len}(x) = \text{len}(x')$, $\text{last}(x) = \text{last}(x')$, and $\text{len}(y) \geq L + \text{len}(x)$, if $\mathcal{F}$ can solve $\{(x,y)\}$ with a loss α using a prompt $\mathcal{P}$ with a length greater than $\mathcal{P} \geq c\text{len}(y)^d$, then $\mathcal{F}$ can also solve $\{(x',y)\}$ with a loss $\alpha + \epsilon$ by using the same prompt $\mathcal{P}$.*

For the loss function $\mathcal{L}_{\text{Nnll}}$, the above result is true for $d > 1$.

Corollary 5.9. *Under $\mathcal{L}_{Nnll}$ loss, Proposition 5.8 holds for any $d > 1$ when other conditions remain unchanged.*

Both Theorem 5.7 and Proposition 5.8 can be used to draw conclusions for multiple samples with long prompts. Here, we present one of them.

Proposition 5.10. *For any transformer $\mathcal{F}$, $\epsilon > 0$, there exist $c > 0, d > 1$ such that for any task $S = \{(x_i, y_i)\}_{i=1}^N$ satisfying $\text{len}(y_i) > \text{len}(x_i)$ and $y_i \neq y_j$, $\text{last}(x_i) = \text{last}(x_j)$ for all $i \neq j \in [N]$, it holds $\mathcal{L}(\mathcal{F}, \mathcal{P} \oplus S) \geq \sum_{i \in \mathbb{Z}_+} \frac{N_i \ln N_i}{N} - \epsilon$ for any prompt $\mathcal{P}$ with a length of more than $c\text{len}(S)^d$, where N_i is the number of queries in S with a length of i.*

As a consequence of Proposition 5.10, if S satisfies $\text{len}(x_i) = \text{len}(x_j)$ for all $i, j \in [N]$, then it is easy to see that $\mathcal{L}(\mathcal{F}, \mathcal{P} \oplus S) \geq \ln N - \epsilon$. Thus, it is not possible to solve such a task with a loss of about $\ln N$ by using a long prompt.

Theorem 5.7 and Proposition 5.10 reveal an important distinction between prompting and training. Training obeys the scaling law (Kaplan et al., 2020; Hoffmann et al., 2022), meaning that more parameters and more training samples lead to better accuracy. On the other hand, Theorem 5.7 and Proposition 5.10 imply that even a very long prompt is not enough to solve multiple tasks with long answers when using a fixed transformer. In other words, **no scaling law exists for hard prompts**.

5.4 Linear prompt

In this section, we will consider linear prompts; that is $d = 1$. In this case, the length of the prompt and that of the answer will remain in a fixed ratio c. We show that the main limitations in Sections 5.2 and 5.3 do not occur in this case, which illustrates certain advantages of linear prompts.

Linear prompt enhances the power of transformers in certain cases. We first show that Theorem 5.4 no longer holds, as demonstrated below.

Proposition 5.11. *For any $\epsilon, c > 0, d = 1$ and $\alpha_1 < \alpha_2$, there exists a transformer $\mathcal{F}$ such that for any given L, there exists a task $S = \{(x, y)\}$ such that $\text{len}(y) > L$ and $\mathcal{F}$ cannot solve S with loss $\alpha_2(L + 1)$ without a prompt, but $\mathcal{F}$ can solve S with loss $\alpha_1(L + 1)$ using a (c, d)-prompt.*

In the case of $\alpha_1 \ll \alpha_2$, this proposition shows that with a $\Omega(\text{len}(y))$ length prompt, the transformer can enhance its ability to solve certain tasks with linear answers.

But when $c \ll 1$, the prompt is also much shorter than the task, and it is easy to see that Proposition 5.6 still holds in this case. Following Proposition 5.6, we have

Corollary 5.12. *For any token set Σ of size T, transformer $\mathcal{F}$, query $x \in \Sigma^*$, and $\alpha, c > 0, d = 1, L \in \mathbb{Z}_+$ satisfying $c < 1/\log_T(T + 1)$ and $L > \frac{\alpha}{1 - c\log_T(T+1)}$, there exists a $y \in \Sigma^*$ with $\text{len}(y) = L$ such that $\mathcal{F}$ cannot solve $\{(x, y)\}$ using an (α, c, d)-prompt.*

Prompt dominating answer phenomenon does not always hold for linear prompts. We show that a linear prompt with a certain c can help alleviate the prompt dominating answer phenomenon.

Proposition 5.13. *For any $\beta > 0, d = 1$, there exist a transformer $\mathcal{F}$ and a constant c such that for any length $L > 0$, there exist queries $x, x' \in \Sigma^*$ and an answer y satisfying $\text{len}(x) = \text{len}(x')$, $\text{last}(x) = \text{last}(x')$, $\text{len}(y) \geq L + \text{len}(x)$ such that $\mathcal{F}$ can solve $\{(x', y)\}$ with certain α by a prompt with length more that $c\text{len}(y)$, but $\mathcal{F}$ cannot solve $\{(x, y)\}$ with $\alpha + \epsilon$ by the same prompt.*

This proposition shows that we can avoid the prompt dominating problem in Theorem 5.7 and Proposition 5.8 by using a linear prompt with a certain transformer and c. For the $\mathcal{L}_{\text{Nnll}}$ loss, we have:

Proposition 5.14. *When considering the $\mathcal{L}_{Nnll}$ loss, Proposition 5.13 still stands.*

Remark 5.15. Please note that Propositions 5.11 and 5.13 only show that a specific transformer can avoid such problems in Theorems 5.4 and 5.7, but not all transformers. This is reasonable because, in certain extreme cases such as $\mathcal{F} \equiv 0$, Propositions 5.11 and 5.13 must not hold.

6 Generalization of prompt

An important issue with prompt engineering is whether prompts for a finite task can be effectively generalized to new samples. In this section, we provide a necessary and sufficient criterion for this kind of generalization. Note that the results in this section hold for arbitrary prompts and are not limited to polynomial ones.

We adopt the usual setting for deriving generalization bounds (Mohri et al., 2018). Let $\mathcal{D}$ be a distribution over $\Sigma^* \times \Sigma^*$ and $S = \{(x_i, y_i)\}_{i=1}^N$ a dataset sampled iid according to $\mathcal{D}$, that is, $S \sim \mathcal{D}^N$. Let

$$\mathcal{A}_{\mathcal{F},\mathcal{P},S,\alpha} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{I}(\mathcal{L}_{\text{nll}}(\mathcal{F}, \mathcal{P} \oplus x_i, y_i) \leq \alpha)$$

be the probability of $\mathcal{P}$ being an α-prompt of $\mathcal{F}$ for S, and

$$\mathcal{A}_{\mathcal{F},\mathcal{P},\mathcal{D},\alpha} = \mathbb{E}_{(x,y)\sim\mathcal{D}}[\mathbf{I}(\mathcal{L}_{\text{nll}}(\mathcal{F}, \mathcal{P} \oplus x, y) \leq \alpha)]$$

be the probability of $\mathcal{P}$ being an α-prompt of $\mathcal{F}$ for the entire data distribution. Then the gap $|\mathcal{A}_{\mathcal{F},\mathcal{P},\mathcal{D},\alpha} - \mathcal{A}_{\mathcal{F},\mathcal{P},S,\alpha}|$ can be used to measure the generalizability of the prompt $\mathcal{P}$. We now provide a uniform generalization bound for prompting.

Theorem 6.1. *For any token set Σ of size T, data distribution $\mathcal{D}$, transformer $\mathcal{F}$, $\alpha > 0$, and $N \in \mathbb{Z}_+$, with a probability $1-\delta$ of $S \sim \mathcal{D}^N$, the following generalization bound holds*

$$|\mathcal{A}_{\mathcal{F},\mathcal{P},\mathcal{D},\alpha} - \mathcal{A}_{\mathcal{F},\mathcal{P},S,\alpha}| \leq \frac{\sqrt{8L\ln(T+1)} + \sqrt{0.5\ln 1/\delta}}{\sqrt{N}},$$

for any prompt $\mathcal{P}$ satisfying $\text{len}(\mathcal{P}) \leq L$.

Proof Idea. We treat the set of $\mathcal{F}(\mathcal{P} \oplus x)$ as a hypothesis space that uses $\mathcal{P}$ as the parameters. Then we can use the generalization bound based on Rademacher complexity (Mohri et al., 2018) on such a hypothesis space to prove the theorem.

This result provides practical guidance for generalization: if the size of the dataset is significantly larger than the length of the prompts, that is $L = o(N)$, then performance on S can be generalized to the entire distribution. We can make this explicit in the following corollary.

Corollary 6.2. *For any $\epsilon \in (0,1)$, if $N \geq (\sqrt{8\text{len}(\mathcal{P})\ln(T+1)} + \sqrt{0.5\ln 1/\delta})^2/\epsilon^2$, then with a probability $1-\delta$ of $S \sim \mathcal{D}^N$, we have $|\mathcal{A}_{\mathcal{F},\mathcal{P},\mathcal{D},\alpha} - \mathcal{A}_{\mathcal{F},\mathcal{P},S,\alpha}| \leq \epsilon$.*

The following theorem shows that the above generalization bound is tight in terms of the main quantities N and $\text{len}(\mathcal{P})$.

Theorem 6.3. *For any token set Σ of size T, $N, L \in \mathbb{Z}_+$, $\alpha \in \mathbb{R}_+$, $\epsilon \in (0,1)$, if $L \geq \frac{N(2-\ln\epsilon)}{\ln T}$, then there exists a transformer $\mathcal{F}$ and a distribution $\mathcal{D}$ such that, with a probability $1-\epsilon$ of $S \sim D^N$, it holds $|\mathcal{A}_{\mathcal{F},\mathcal{P},\mathcal{D},\alpha} - \mathcal{A}_{\mathcal{F},\mathcal{P},S,\alpha}| \geq 1-\epsilon$ for some prompt $\mathcal{P}$ satisfying $\text{len}(\mathcal{P}) = L$.*

Proof Idea. The distribution $\mathcal{D}$ is defined on a finite set $\mathcal{S}$ with more than $[N/\epsilon]$ elements. Then every $S_k \sim \mathcal{D}^N$ has a distinct prompt $\mathcal{P}_k$ with the given length. Using the theory of memorization (Mahdavi et al., 2023; Yu et al., 2025), we construct an $\mathcal{F}$ that can solve $\mathcal{P}_k \oplus S_k$ with loss α but cannot solve $\mathcal{P}_k \oplus (\mathcal{S} \setminus S_k)$ with loss α. Now, the theorem derives from the fact $|S_k|/|\mathcal{S}| < \epsilon$.

Theorem 6.3 shows that, for prompts with a length greater than $\Omega(N)$, their effect on S may not generalize to $\mathcal{D}$. By Corollary 6.2, a sufficient condition for such a generalization is $N \geq \overline{O}(L)$. **We thus obtain a tight bound:**

Corollary 6.4. *A necessary and sufficient condition for the performance of $\mathcal{P}$ on $S \sim \mathcal{D}^N$ to generalize on $\mathcal{D}$ for any transformer and distribution $\mathcal{D}$ is $N \geq \overline{O}(\text{len}(\mathcal{P}))$, when omitting other small quantities.*

7 Conclusion

This work establishes the first systematic theoretical framework for hard (discrete) prompts in LLMs. We settle three key theoretical problems: computational complexity, inherent expressiveness limits, and tight generalization bounds. Our results formally explain why short prompts fail for long tasks and why long prompts cause answer domination. These findings unify empirical phenomena in prompt engineering and provide provable guidance for practical LLM deployment.

Limitation and future work. While the results in Sections 4 and 6 are quite complete, the results in Section 5 mainly discuss the limitations of hard prompts. While limitations are indeed significant and informative, an equally important yet challenging task is to establish sufficient conditions and develop construction methods for hard prompts to enhance the capability of transformers.

STATEMENTS

AI USE STATEMENT

LLMs were used for textual revision of the manuscript and did not contribute to generating the paper's main contribution. The author is responsible for all the contents of this paper.

ETHICS STATEMENT

Our research is grounded in theoretical foundations, focusing on the interpretability of prompt engineering, and it does not raise any ethical or moral concerns.

REFERENCES

Zhangir Azerbayev, Hailey Schoelkopf, Keiran Paster, Marco Dos Santos, Stephen McAleer, Albert Q. Jiang, Jia Deng, Stella Biderman, and Sean Welleck. Llemma: An open language model for mathematics. In *ICLR*, 2024.

Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. A neural probabilistic language model. In *Advances in Neural Information Processing Systems 13*, pp. 932–938, 2000.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michał Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, and Torsten Hoefler. Graph of thoughts: solving elaborate problems with large language models. In *AAAI'24*, pp. 17682–17690, 2024.

Aman Bhargava, Cameron Witkowski, Shi-Zhuo Looi, and Matt Thomson. What's the magic word? a control theory of llm prompting. *arXiv preprint arXiv:2310.04444*, 2023.

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

Tonmoy Debnath, Md Nurul Absar Siddiky, Muhammad Enayetur Rahman, Prosenjit Das, and Antu Kumar Guha. A comprehensive survey of prompt engineering techniques in large language models. *TechRxiv, techrxiv.174140719.96375390/v1*, 2025.

Jessica Ficler and Yoav Goldberg. Controlling linguistic style aspects in neural language generation. In *Proceedings of the Workshop on Stylistic Variation*, pp. 94 – 104, 2017.

Tianyu Gao, Adam Fisch, and Danqi Chen. Making pre-trained language models better few-shot learners. In *Proceedings of ACL-IJCNLP*, pp. 3816–3830, 2021.

Roee Hendel, Mor Geva, and Amir Globerson. In-context learning creates task vectors. *arXiv preprint arXiv:2310.15916*, 2023.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katie Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Jack W. Rae, Oriol Vinyals, and L. Sifre. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

Jerry Yao-Chieh Hu, Wei-Po Wang, Ammar Gilani, Chenyang Li, Zhao Song, and Han Liu. Fundamental limits of prompt tuning transformers: Universality, capacity and efficiency. In *ICLR'25*, 2025.

Hong Jun Jeon, Jason D Lee, Qi Lei, and Benjamin Van Roy. An information-theoretic analysis of in-context learning. *arXiv preprint arXiv:2401.15530*, 2024.

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

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33: 9459–9474, 2020.

Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*, 2021.

Zongqian Li, Yixuan Su, and Nigel Collier. A survey on prompt tuning. In *3rd Workshop on Efficient Systems for Foundation Models (ES-FoMo) at ICML*, 2024. URL `https://openreview.net/pdf?id=JEMGDajQ1G`. OpenReview preprint.

Zongqian Li, Yinhong Liu, Yixuan Su, and Nigel Collier. Prompt compression for large language models: A survey. In *Proceedings of the 2025 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 7182–7195, 2025.

Yong Lin, Shange Tang, Bohan Lyu, Jiayun Wu, Hongzhou Lin, Kaiyu Yang, Jia Li, Mengzhou Xia, Danqi Chen, Sanjeev Arora, and Chi Jin. Goedel-prover: A frontier model for open-source automated theorem proving. In *COLM*, 2025.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. Gpt understands, too. *arXiv preprint arXiv:2103.10385*, 2021.

Sadegh Mahdavi, Renjie Liao, and Christos Thrampoulidis. Memorization capacity of multi-head attention in transformers. *arXiv preprint arXiv:2306.02010*, 2023.

William Merrill and Ashish Sabharwal. The expressive power of transformers with chain of thought. In *ICLR 2024*, 2024.

Maxime Meyer, Mario Michelessa, Caroline Chaux, and Vincent YF Tan. Memory limitations of prompt tuning in transformers. *arXiv preprint arXiv:2509.00421*, 2025.

Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2018.

Ryumei Nakada, Wenlong Ji, Tianxi Cai, James Zou, and Linjun Zhang. A theoretical framework for prompt engineering: Approximating smooth functions with transformer prompts. *arXiv preprint arXiv:2503.20561*, 2025.

Jorge Pérez, Pablo Barceló, and Javier Marinkovic. Attention is turing-complete. *Journal of Machine Learning Research*, 22(75):1–35, 2021.

Aleksandar Petrov, Philip HS Torr, and Adel Bibi. Prompting a pretrained transformer can be a universal approximator. *arXiv preprint arXiv:2402.14753*, 2024.

Archiki Prasad, Peter Hase, Xiang Zhou, and Mohit Bansal. Grips: Gradient-free, edit-based instruction search for prompting large language models. *arXiv preprint arXiv:2203.07281*, 2022.

Reid Pryzant, Dan Iter, Jerry Li, Yin Tat Lee, Chenguang Zhu, and Michael Zeng. Automatic prompt optimization with "gradient descent" and beam search. *arXiv preprint arXiv:2305.03495*, 2023.

Guanghui Qin and Jason Eisner. Learning how to ask: Querying lms with mixtures of soft prompts. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2021.

Ruizhong Qiu, Zhe Xu, Wenxuan Bao, and Hanghang Tong. Ask, and it shall be given: On the turing completeness of prompting. In *ICLR*, 2025.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Antonio Sabbatella, Andrea Ponti, Ilaria Giordani, Antonio Candelieri, and Francesco Archetti. Prompt optimization in large language models. *Mathematics*, 12(929), 2024.

Timo Schick and Hinrich Schütze. It's not just size that matters: Small language models are also few-shot learners. *arXiv preprint arXiv:2009.07118*, 2020.

Wei Shen, Ruida Zhou, Jing Yang, and Cong Shen. On the training convergence of transformers for in-context classification of gaussian mixtures. *arXiv preprint arXiv:2410.11778*, 2024.

Taylor Shin, Yasaman Razeghi, Robert L. Logan, Eric Wallace, and Sameer Singh. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. *arXiv preprint arXiv:2010.15980*, 2020.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Tianxiang Sun, Yunfan Shao, Hong Qian, Xuanjing Huang, and Xipeng Qiu. Black-box tuning for language-model-as-a-service. In *International Conference on Machine Learning*, pp. 20841–20855. PMLR, 2022.

Haiyu Wang and Yuanyuan Lin. Prompt tuning transformers for data memorization. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Yihan Wang, Jatin Chauhan, Wei Wang, and Cho-Jui Hsieh. Universality and limitations of prompt tuning. *Advances in Neural Information Processing Systems*, 36:75623–75643, 2023.

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

Yuxin Wen, Neel Jain, John Kirchenbauer, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Hard prompts made easy: Gradient-based discrete optimization for prompt tuning and discovery. *Advances in Neural Information Processing Systems*, 36:51008–51025, 2023.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. An explanation of in-context learning as implicit bayesian inference. *arXiv preprint arXiv:2111.02080*, 2021.

Tong Yang, Yu Huang, Yingbin Liang, and Yuejie Chi. In-context learning with representations: Contextual generalization of trained transformers. *Advances in Neural Information Processing Systems*, 37:85867–85898, 2024.

Zijian Yang, Yingbo Gao, Weiyue Wang, and Hermann Ney. Predicting and using target length in neural machine translation. In *Asia-Pacific Conference on Language and Information Processing (AACL)*, pp. 389 – 395, 2020.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik R Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *NeurIPS'23*, 2023.

Lijia Yu, Xiao-Shan Gao, and Lijun Zhang. Analyzing the power of chain of thought through memorization capabilities. In *PeurIPS*, 2025.

Tianjun Zhang, Xuezhi Wang, Denny Zhou, Dale Schuurmans, and Joseph E Gonzalez. Tempera: Test-time prompting via reinforcement learning. *arXiv preprint arXiv:2211.11890*, 2022.

A Details of Setting

Notation. In this paper, we use $O(A)$ to mean a value not greater than cA for some constant c, and $\overline{O}$ to mean that small quantities, such as logarithms, are omitted. We use $\Omega(A)$ to mean a value not less than cA for some constant c.

Positional Encoding. *Relative position embedding* is used: $R(xQ_i, K_i x^\tau) \in \mathbb{R}^{n\times n}$, where the (u,v)-th weight of $R(xQ_i, K_i x^\tau)$ is $x[u]Q_i A_{u-v} K_i x^\tau[v]$, where $x[i]$ is the i-th row of x and $A_{u-v} \in [-1,1]^{W\times W}$ is the matrix for the relative position embedding. Note that A_{u-v} is free of i. This definition aligns with most relative position embeddings, such as RoPE (Su et al., 2024).

B Proofs in Section 4

B.1 Scope of application of the conclusion

Our proof relies on standard complexity-theoretic techniques, specifically by constructing a reduction from an NP-complete problem to the target problem.

Our proof is applicable to the hypothesis space we defined and all hypothesis spaces containing the transformers constructed in our proof. For other hypothetical spaces $\mathbb{H}$, if for any $\mathcal{F}$ used in the proof, there is a $\mathcal{F}_h \in \mathbb{H}$ such that $\mathcal{F}_h = \mathcal{F}$ on the constructed point, and $\mathcal{F}_h$ has at most polynomial multiples scale of $\mathcal{F}$, then our proof is applicable on $\mathbb{H}$.

This point is not strict, because in order to complete any task by transformer, we need the transformer class to have adequate expressive capacity. And considering that we only use a single attention layer transformer with single structure in our proof, so it is not hard to approximate it.

On the other hand, our results 4.5 and B.4 are strictly applicable to globally optimal prompt strategies. Since most gradient-based prompt strategies in practice merely converge to a local optimum, our conclusions do not extend to such empirically derived locally optimal strategies.

B.2 Proof of Theorem 4.2

Definition B.1 (The 3-SAT decision problem)**.** Let $\{z_i\}_{i=1}^N$ be Boolean variables and $\Phi = \wedge_{j=1}^M \phi_j$, where $\phi_j = \vee_{k=1,2,3} z'_{j_k}$ is a Boolean clause and $z'_{j_k} \in \{z_i\}_{i=1}^n \cup \{\neg z_i\}_{i=1}^n$. The 3-SAT problem is to decide whether there exist values for the variables under which $\Phi = 1$.

Now we prove Theorem 4.2 and Corollary 4.3 together.

Proof. Firstly, for a given prompt $\mathcal{P}$, it is easy to check whether $\mathcal{P}$ is a solution such that $\mathbf{Prompt}(\mathcal{F}, S, \alpha, c, d) = 1$ in polynomial time based on the size of Σ, S, α, c, d and the parameters of $\mathcal{F}$. Therefore, $\mathbf{Prompt}$ is an NP problem.

To prove Theorem 4.2, it suffices to show that an NP-complete problem (here, we use 3-SAT) can be reduced to $\mathbf{Prompt}(\mathcal{F}, S, \alpha, c, d) = 1$.

Let $\Phi = \wedge_{j=1}^M \phi_j$ be a 3-SAT problem as defined in Definition B.1. Without loss of generality, we need Φ to satisfy the conditions that: $N, M \geq 6$ and each z_i is in at least one of ϕ_j, which implies $N \leq 3M$. It is obvious that all SAT problems can be reduced to this kind of situation.

We first construct a Σ, a dataset S and a transformer $\mathcal{F}$ based on Φ.

Step 1. Construction of Σ, $\mathcal{F}$, S and α, c, d.

To prove the (1) in Corollary 4.3 together, we will consider a situation in which $|S| = 1$.

The symbol set is $\Sigma = \{\sigma_i\}_{i=1}^{N+M}$ and the dataset is $S = \{(\sigma_{1+N}, \sigma_{2+N} \oplus \sigma_{3+N} \oplus \cdots \oplus \sigma_{M+N})\} \subset \Sigma^* \times \Sigma^*$, where the query set $\{\sigma_{1+N}\}$ only contains one sequence with a single symbol σ_{1+N} in it.

Let $\alpha = \ln 2$, $c \geq 3N$, and $d \geq 1$. The transformer is defined as follows.

(1) The embedding layer: Each σ_i will be embedded into a $4(N+M)$-dimensional vector v_{σ_i}, such that for the $i \in [1, N+M]$, the $(4i-3)$-th to $4i$-th weights in the v_{σ_i} are 1, and other weights are 0.

For a $x \in \Sigma^*$, let $v(x)$ be its embedding matrix. And the position matrix A_i is an identity matrix I for any $i \in N_+$.

(2) The hidden layers: In the first hidden layer $\mathcal{F}_1(x)$, the attention layer is defined as: the position embedding is I and $Q_1K_1 = I$, $V_1 = I$. The feedforward layer $\text{FNN}_1(x)$ in the first layer is defined as $\text{FNN}_1(x) = 0$.

Then, based on the definition of the embedding layer, the output of the first layer when the input is x is $\mathcal{F}_1(x) = v(x) + \text{softmax}(4I_{\text{len}(x)} + M)v(x)$. Hence, its last row is $v_{x[\text{len}(x)]} + \sum_{i=1}^{\text{len}(x)} \frac{v_{x[i]}}{\text{len}(x)}$. Name it as $N(x)$, and let $N_i(x)$ be the i-th weight of $N(x)$. Before defining the next layer, we first define 3 kinds of Relu networks $G_i(x) : \mathbb{R} \to \mathbb{R}$:

$G_1(x) = \text{Relu}(6N^2(x - 1/N)) + x$; and $G_2(x) = x + \text{Relu}((N + M + 3)(1/(N + M + 2) - x))$, $\mathcal{G}_3(x) = \text{Relu}(-x + 1/(N + M + 2)) + x$.

Then we define the next hidden layer, which is also the last hidden layer. Let the parameters in the attention layer all be 0, and the FNN layer makes the output of the last hidden layer $\mathcal{F}_l(x)$ to be:

$$\begin{aligned}\mathcal{F}_l(x) = \quad & (N_4(x), \mathcal{G}_1(N_4(x)), \mathcal{G}_2(N_4(x)), \mathcal{G}_3(N_4(x)),\\ & N_8(x), \mathcal{G}_1(N_8(x)), \mathcal{G}_2(N_8(x)), \mathcal{G}_3(N_8(x)),\\ & \ldots, N_{4(N+M)}(x), \mathcal{G}_1(N_{4(N+M)}(x)), \mathcal{G}_2(N_{4(N+M)}(x)), \mathcal{G}_3(N_{4(N+M)}(x))).\end{aligned}$$

Notice that in the $N(x)$, there are $N_{4i-3}(x) = N_{4i-2}(x) = N_{4i-1}(x) = N_{4i}(x) = \frac{\text{Num}_{\sigma_i}(x)}{\text{len}(x)} + I(x[\text{len}(x)] = \sigma_i)$, so it is easy to implement the above output by FNN layer. In this context, $\text{Num}_{\sigma_i}(x)$ is the number of σ_i in the sentence x.

(3) The output layer: The first $N + 1$ elements of the transformer output are 0. For $j \in [M]$, based on the different forms of ϕ_j, the $(N + 1 + j)$-th element of the transformer output has a different form (3.1)-(3.4), which is defined below (for convenience, let the $(N + M + 1)$-th element of the transformer be the probability for σ_0).

(3.1) $M(\mathcal{G}_1(N_{4i'}(x)) + \mathcal{G}_1(N_{4j'}(x)) + \mathcal{G}_1(N_{4k'}(x)) - 3/N)$

$-100M^2 \sum_{i\in[N+1,M+N],i\neq N+j,} N_{4i}(x)$ if $\phi_j = z_{i'} \cup z_{j'} \cup z_{k'}$ for some $i', j', k' \in [N]$;

(3.2) $M(\mathcal{G}_1(N_{4i'}(x)) + \mathcal{G}_1(N_{4j'}(x)) - 2.1(1/N - \mathcal{G}_2(N_{4k'}(x)) + \mathcal{G}_3(N_{4k'}(x))) + 0.1/N) - 100M^2 \sum_{i\in[N+1,M+N],i\neq N+j} N_{4i}(x)$ if $\phi_j = z_{i'} \cup z_{j'} \cup \neg z_{k'}$ for some $i', j', k' \in [N]$;

(3.3) $M(\mathcal{G}_1(N_{4i'}(x)) - 2.1(1/N - \mathcal{G}_2(N_{4j'}(x)) + \mathcal{G}_3(N_{4j'}(x))) - 2.1(1/N - \mathcal{G}_2(N_{4k'}(x)) + \mathcal{G}_3(N_{4k'}(x))) + 0.1/N) - 100M^2 \sum_{i\in[N+1,M+N],i\neq N+j} N_{4i}(x)$ if $\phi_j = z_{i'} \cup \neg z_{j'} \cup \neg z_{k'}$ for some $i', j', k' \in [N]$;

(3.4) $M(6.2/N - 2.1(1/N - \mathcal{G}_2(N_{4i'}(x)) + \mathcal{G}_3(N_{4i'}(x))) - 2.1(1/N - \mathcal{G}_2(N_{4j'}(x)) + \mathcal{G}_3(N_{4j'}(x))) - 2.1(1/N - \mathcal{G}_2(N_{4k'}(x)) + \mathcal{G}_3(N_{4k'}(x)))) - 100M^2 \sum_{i\in[N+1,M+N],i\neq N+j} N_{4i}(x)$ if $\phi_j = \neg z_{i'} \cup \neg z_{j'} \cup \neg z_{k'}$ for some $i', j', k' \in [N]$.

It can be easily verified that this transformer can indeed be realized with $O(\text{poly}(M, N))$ parameters and requires $O(\log(\text{poly}(M, N)))$ bits for each parameter value.

Step 2. We show that if $\mathbf{Prompt}(\mathcal{F}, S, \alpha, c, d))$ is decidable, then the 3-SAT problem Φ is also decidable.

To be convenient, write $2.1(1/N - \mathcal{G}_2(x) + \mathcal{G}_3(x)) = 2.1(1/N - (N + M + 2)Relu(1/(N + M + 2) - x)) = G_4(x)$. The proof for this step has three parts.

Part 1. If $\mathbf{Prompt}(\mathcal{F}, S, \alpha, c, d) = 1$, then the 3-SAT problem Φ has a solution.

Assume that **Prompt** problem has a solution $\mathcal{P}$, then we can find the following solution for the 3-SAT problem Φ: $z_i = 1$ if $\text{Num}_{\sigma_i}(\mathcal{P})/(\text{len}(\mathcal{P}) + M) \geq 1/N$ where $i \in [N]$; $z_i = 0$ if $\text{Num}_{\sigma_i}(\mathcal{P})/(\text{len}(\mathcal{P}) + 1) \leq 1/(N + M + 2)$ where $i \in [N]$; in other cases, z_i can be taken arbitrarily. Please note that this value will not create ambiguity.

We will show why this is a solution to the 3-SAT problem Φ. To demonstrate that, we only need to show that each $\phi_j = 1$ holds for any $j \in [M]$ in Φ. For convenience, in this part, we write $\mathcal{P} \oplus \sigma_{1+N} \oplus \sigma_{2+N} \oplus \sigma_{3+N} \oplus \cdots \oplus \sigma_{j+N} = x_j$, where $j \in [M]$.

Firstly, for any i, the probability of $\pi_{\mathcal{F}}(\sigma_{i+1+N} \mid x_i)$ must be higher than the probability of $\pi_{\mathcal{F}}(\sigma_j \mid x_i)$ for any $j \neq i+1+N$.

If not, there must be $\mathcal{L}_{\text{nll}}(\mathcal{F}, x_i, \sigma_{N+1+i}) > \ln 2$, then $L(\mathcal{F}, S) > \ln 2 = \alpha$, which is contradictory to conclusion that $\mathbf{Prompt}(\mathcal{F}, S, \alpha, c, d) = 1$.

Secondly, because the probability of $\pi_{\mathcal{F}}(\sigma_{i+1+N} \mid x_i)$ is the largest, we have that:

(1) if $\phi_i = z_{i'} \cup z_{j'} \cup z_{k'}$ for some $i', j', k' \in [N]$, consider the definition of $\mathcal{P}$ and transformer $\mathcal{F}$. There must be $\mathcal{G}_1(N_{4i'}(x_i)) + \mathcal{G}_1(N_{4j'}(x_i)) + \mathcal{G}_1(N_{4k'}(x_i)) - 3/N > 0$; consider that $\mathcal{G}_1$ is increased with input and $\mathcal{G}_1(1/N) = 1/N$. so at least one of $N_{4i'}(x_i) \geq 1/N$, $N_{4j'}(x_i) \geq 1/N$, and $N_{4k'}(x_i) \geq 1/N$ stands, that is $\text{Num}_{\sigma_i}(\mathcal{P})/(\text{len}(\mathcal{P}) + i) \geq 1/N$, which implies $z_{i'} = 1$ or $z_{j'} = 1$ or $z_{k'} = 1$. Thus, we have that $\phi_i = 1$;

(2) if $\phi_i = z_{i'} \cup z_{j'} \cup \neg z_{k'}$ for some $i', j', k' \in [N]$, then we have that $\mathcal{G}_1(N_{4i'}(x_i)) + \mathcal{G}_1(N_{4j'}(x_i)) - \mathcal{G}_4(N_{4k'}(x_i)) + 0.1/N > 0$. If $\neg z_{k'} = 0$, then $N_{4k'}(x_i) \geq 1/(N+M+2)$ based on the definition of solution; there are $-\mathcal{G}_4(N_{4k'}(x_i)) + 0.1/N = -2/N$ based on the definition of $\mathcal{G}_2$ and $\mathcal{G}_3$, which implies $\mathcal{G}_1(N_{4i'}(x_i)) + \mathcal{G}_1(N_{4j'}(x_i)) > 2/N$. similar to before, at least one of $N_{4i'} \geq 1/N$ and $N_{4j'} \geq 1/N$ stands. So $\neg z_{k'} = 0$ implies $z_{i'} = 1$ or $z_{j'} = 1$, hence $\phi_i = 1$; when $\neg z_{k'} = 1$, there is also $\phi_i = 1$;

(3) if $\phi_i = z_{i'} \cup \neg z_{j'} \cup \neg z_{k'}$ for some $i', j', k' \in [N]$, then we have that $\mathcal{G}_1(N_{4i'}(x_i)) - \mathcal{G}_4(N_{4j'}(x_i)) - \mathcal{G}_4(N_{4k'}(x_i)) + 0.1/N > 0$. Similarly, if $\neg z_{k'} = 0$ and $\neg z_{j'} = 0$, then $N_{4j'}(x_i), N_{4k'}(x_i) \geq 1/(N+M+2)$; hence $-\mathcal{G}_4(N_{4j'}(x_i)) - \mathcal{G}_4(N_{4k'}(x_i)) + 0.1/N \leq -1/N$, which implies that $N_{4i'}(x_i) \geq 1/N$; so $\neg z_{k'} = 0$ and $\neg z_{j'} = 0$ imply $z_{i'} = 1$; hence $\phi_i = 1$; when $\neg z_{k'} = 1$ or $\neg z_{j'} = 1$, this is also $\phi_i = 1$;

(4) if $\phi_i = \neg z_{i'} \cup \neg z_{j'} \cup \neg z_{k'}$ for some $i', j', k' \in [N]$, then we have that $6.2/N - \mathcal{G}_4(N_{4i'}(x_i)) - \mathcal{G}_4(N_{4j'}(x_i)) - \mathcal{G}_4(N_{4k'}(x_i)) > 0$. Consider that when $\neg z_{i'} = 0$, there is $N_{4i'}(x_{i-1}) \geq 1/(N+M+2)$; then $2.1(1/N - \mathcal{G}_4(N_{4i'}(x_i))) = 2.1/N$, similar for j', k'. So, when $\neg z_{i'} = 0, \neg z_{j'} = 0, \neg z_{k'} = 0$, there is $6.2/N - \mathcal{G}_4(N_{4i'}(x_i)) - \mathcal{G}_4(N_{4j'}(x_i)) - \mathcal{G}_4(N_{4k'}(x_i)) \leq 0$, which is contradictory to the above result. Therefore, at least one of $\neg z_{i'} = 1$, $\neg z_{j'} = 1$, or $\neg z_{k'} = 1$ stands, so $\phi_i = 1$.

Thus, we prove Part One.

Part 2. If $\mathbf{Prompt}(\mathcal{F}, S, \alpha, c, d) = 0$, then Φ has no solution, or it has a solution $z_i = 1$ for all i or $z_i = 0$ for all i.

We just need to show that if the 3-SAT problem Φ has a solution that is not $z_i = 1$ for all i or $z_i = 0$ for all i, then such $\mathbf{Prompt} = 1$.

Let σ_i^{3M} denote the sequence that contains only $3M$ number of σ_i. Then let $\mathcal{P} = \sigma_{i_1}^{3M} \oplus \sigma_{i_2}^{3M} \oplus \ldots \sigma_{i_{j-1}}^{3M} \oplus \sigma_{i_j}^{3M}$ where i_j satisfies $z_{i_j} = 1$ in the solution of the 3-SAT problem Φ. It is easy to see that $\text{len}(\mathcal{P}) \leq c\text{len}(y)^d$ by the value of c, d. For convenience, in this part, we still write $\mathcal{P} \oplus \sigma_{1+N} \oplus \sigma_{2+N} \oplus \sigma_{3+N} \oplus \cdots \oplus \sigma_{j+N} = x_j$, where $j \in [M]$; we just need to show that $-\ln \pi_{\mathcal{F}}(\sigma_{1+N+i} | x_i) \leq \ln 2/(M+1)$ for any $i \in [M]$.

Because the solution is not $z_i = 1$ for all i or $z_i = 0$ for all i, we know that $\frac{N_{\sigma_i}(x_k)}{\text{len}(x_k)} \geq \frac{3M}{3M(N-1)+1+M} \geq 1/(N-0.5)$ for any $i, k \in [M]$ such that $z_i = 1$ in the solution of Φ, and $\frac{N_{\sigma_i}(x_k)}{\text{len}(x_k)} = 0$ when $z_i = 0$ in the solution of Φ.

Then we consider the $\mathcal{F}(x_q)$ where $q \in [M]$, such that:

(1) if $\phi_q = z_i \cup z_j \cup z_k$ for some $i, j, k \in [N]$, then at least one of $N_{4i}(x_q) \geq 1/(N-0.5)$, $N_{4j}(x_q) \geq 1/(N-0.5)$, and $N_{4k}(x_{q-1}) \geq 1/(N-0.5)$ stands, so

$$\begin{aligned} & M(\mathcal{G}_1(N_{4i}(x_{q-1})) + \mathcal{G}_1(N_{4j}(x_{q-1})) + \mathcal{G}_1(N_{4k}(x_{q-1})) - 3/N) \\ & -100M^2 \textstyle\sum_{t \in [N+1, M+N], t \neq N+q} N_{4t}(x_{q-1}) \\ = \; & M(\mathcal{G}_1(N_{4i}(x_{q-1})) + \mathcal{G}_1(N_{4j}(x_{q-1})) + \mathcal{G}_1(N_{4k}(x_{q-1})) - 3/N) \\ \geq \; & M(3 - 3/N) > M. \end{aligned}$$

We use $\mathcal{G}_1(1/(N-0.5)) > 3$ and $\mathcal{G}_1(x) \geq 0$ when $x \geq 0$ here.

On the other hand, for $t \neq q$, there is $-100M^2 \sum_{i \neq N+t, i \in [N+1, M+N]} N_{4i}(x_q) \leq -100M^2$ because $N_{4(N+q)}(x_q) \geq 1$, so the $N+t$ weight has a value of no more than 0. Consider that the value of the first N+1-weight is 0, so $-\ln \pi_{\mathcal{F}}(\sigma_{q+1+N} \mid x_{q-1}) \leq -\ln \frac{e^M}{e^M+M+N} \leq \ln(1+4M/e^M) \leq 4M/e^M \leq \ln 2/(M+1)$; use $M \geq 6$ and $N \leq 3M$ here, which is what we want;

(2) If $\phi_q = z_i \cup z_j \cup \neg z_k$ for some $i, j, k \in [N]$, then at least one of $N_{4i}(x_q) \geq 1/(N-0.5)$, $N_{4j}(x_q) \geq 1/(N-0.5)$ and $N_{4k}(x_q) = 0$ stands, so

$$\begin{array}{rl} & M(\mathcal{G}_1(N_{4i}(x_q)) + \mathcal{G}_1(N_{4j}(x_q)) - \mathcal{G}_4(N_{4k}(x_q)) + 0.1/N) \\ & -100M^2 \sum_{t \in [N+1, M+N], t \neq N+q} N_{4t}(x_q) \\ = & M(\mathcal{G}_1(N_{4i}(x_q)) + \mathcal{G}_1(N_{4j}(x_q)) - \mathcal{G}_4(N_{4k}(x_q)) + 0.1/N) \\ \geq & M \min\{2.1 - 2/N, 3 - 2/N\} > M. \end{array}$$

where we use $\mathcal{G}_1(1/(N-0.5)) > 3$ and $G_4(0) = -2.1(1-1/N)$. Similar to before, other weights have values of no more than 0, so $-\ln \pi_{\mathcal{F}}(\sigma_{q+1+N} \mid x_q) \leq -\ln \frac{e^M}{e^M+M+N} \leq \ln 2/(M+1)$;

(3) if $\phi_q = z_i \cup \neg z_j \cup \neg z_k$ for some $i, j, k \in [N]$, then at least one of $N_{4i}(x_q) \geq 1/(N-0.5)$, $N_{4j}(x_q) = 0$ and $N_{4k}(x_q) = 0$ stands, so

$$\begin{array}{rl} & M(\mathcal{G}_1(N_{4i}(x_q)) - \mathcal{G}_4(N_{4j}(x_q)) - \mathcal{G}_4(N_{4k}(x_q)) + 0.1/N) \\ & -100M^2 \sum_{t \in [N+1, M+N], t \neq N+q} N_{4t}(x_q) \\ = & M(\mathcal{G}_1(N_{4i}(x)) - \mathcal{G}_4(N_{4j}(x)) - \mathcal{G}_4(N_{4k}(x)) + 0.1/N) \\ \geq & M \min\{3 - 4.1/N, 2.1 - 4.1/N\} > M. \end{array}$$

Similar to before, other weights have a value of no more than 0. Therefore, $-\ln \pi_{\mathcal{F}}(\sigma_{q+1+N} \mid x_q) \leq -\ln \frac{e^M}{e^M+M+N} \leq \ln 2/(M+1)$;

(4) If $\phi_1 = \neg z_i \cup \neg z_j \cup \neg z_k$ for some $i, j, k \in [N]$, then we have at least one of $N_{4i}(x_q) = 0$, $N_{4j}(x_q) = 0$, and $N_{4k}(x_q) = 0$ standing, so

$$\begin{array}{rl} & M(6.2/N - \mathcal{G}_4(N_{4i}(x_q)) - \mathcal{G}_4(N_{4j}(x_q)) - \mathcal{G}_4(N_{4k}(x_q))) \\ & -100M^2 \sum_{t \in [N+1, M+N], t \neq N+q} N_{4t}(x_q) \\ = & M(6.2/N - \mathcal{G}_4(N_{4i}(x_q)) - \mathcal{G}_4(N_{4j}(x_q)) - \mathcal{G}_4(N_{4k}(x_q))) \\ \geq & M(2.1 - 0.1/N) > M. \end{array}$$

Similar to before, other weights have a value of no more than 0, so $-\ln \pi_{\mathcal{F}}(\sigma_{q+1+N} \mid x_q) \leq -\ln \frac{e^M}{e^M+M+N} \leq \ln 2/(M+1)$.

Therefore, we have that $\mathcal{L}(\mathcal{F}, S) \leq (M+1)\ln 2/(M+1) = \ln 2$, so we prove Part Two.

Part 3. Now we prove the theorem. If we can solve the $\mathbf{Prompt}$ problem, then we can solve the 3-SAT existence problem as follows:

(1) Construct such transformer $\mathcal{F}$, dataset S and symbols Σ and c, α, e based on Φ as before;

(2) If $\mathbf{Prompt}(\mathcal{F}, S, \alpha, c, d) = 1$, then the 3-SAT Problem Φ has a solution;

(3) If $\mathbf{Prompt}(\mathcal{F}, S, \alpha, c, d) = 0$, then the 3-SAT Φ problem has no solution, or the solution satisfies that all the values of z_i are the same, which is easy to check in polynomial time. □

B.3 Computational complexity of prompting for greedy sampling

Greedy sampling is used in many works on prompting (Wang & Lin, 2025; Qiu et al., 2025) as well as in theoretical studies of transformers (Pérez et al., 2021; Merrill & Sabharwal, 2024; Yu et al., 2025). In this case, the output for an autoregressive transformer is computed as follows. Let the initial value $y = ()$.

(1) Let s be the token that maximizes $\pi_{\mathcal{F}}(s \mid x \oplus y)$.

(2) If $s \neq \sigma_0$, insert s as the last element of y, and return to step (1).

(3) If $s = \sigma_0$, stop and return the output $y = \overline{\mathcal{F}}(x)$.

Under this type of output, we have the following:

Prompt. We say that a transformer $\mathcal{F}$ *solves the task* $S = \{(x_i, y_i)\}_{i=1}^N$ *with prompt* $\mathcal{P}$, if $y_i = \overline{\mathcal{F}}(\mathcal{P} \oplus x_i)$ for $i = 1, \ldots, N$, or equivalently, $\mathcal{F}$ is a memorization network for $\mathcal{P} \oplus S$.

Determining the existence of prompts is also NPC.

Proposition B.2. *For given $c, d > 0$, $\mathcal{F}$, and task $S = \{(x_i, y_i)\}_{i=1}^N$, deciding whether there exists a prompt $\mathcal{P} \in \Sigma^*$ such that $y_i = \overline{\mathcal{F}}(\mathcal{P} \oplus x_i)$ and $\mathrm{len}(\mathcal{P}) \leq c\mathrm{len}(S_a)^d$ is NPC. The result is true for $N = 1$.*

The proof is similar to that for Theorem 4.2; simply ignore the part regarding loss calculation (the first part in step 2 of part 1 and the calculation of $-\ln(\pi_{\mathcal{F}}(\sigma_{1+N+i}|x_i))$ in part 2).

We can similarly define the optimization problem to find a prompt.

Definition B.3. The problem $\mathrm{DPO}_d(\mathcal{F}, S, c, d)$ is defined as follows: for the given transformer $\mathcal{F}$, task S, and $c, d \in \mathbb{R}_{>0}$, the problem DPO_d is to solve the following optimization problem to obtain a prompt $\mathcal{P}$:

$$\mathrm{argmax}_{\mathcal{P}\in\Sigma^*, \mathrm{len}(\mathcal{P})\leq c\,\mathrm{len}(S_a)^d} \sum_{(x_i,y_i)\in S} \mathbf{I}(\overline{\mathcal{F}}(\mathcal{P} \oplus x_i) = y_i).$$

Then, by Corollary B.2, we have that:

Corollary B.4. $\mathrm{DPO}_d(\mathcal{F}, S, c, d)$ *is an NP-hard problem.*

C Proofs of Section 5.1

C.1 Proof of Proposition 5.1

We use the following result.

Theorem C.1 (Theorem 4.4 of (Yu et al., 2025)). *Let $S = \{(x_i, y_i)\}_{i=1}^N \subset \Sigma^* \times \Sigma^*$ satisfy that $x_i[1 : \mathrm{len}(x_j)] \neq x_j$ for any $i, j \in [N]$ such that $\mathrm{len}(x_i) \geq \mathrm{len}(x_j)$, then there is a transformer $\mathcal{F}$ that can memory S.*

In the setting of (Yu et al., 2025), $\mathcal{F}$ can memory S means that: $\pi_{\mathcal{F}}(y_i[k]|x_i \oplus y_i[1 : k-1]) > \pi_{\mathcal{F}}(\sigma_j|x_i \oplus y_i[1 : k-1])$ for any $\sigma_j \neq y_i[k]$ and $i \in [N]$, $k \in [\mathrm{len}(y_i)]$. Therefore, we can prove Proposition 5.1 as follows.

Proof. For length $l \leq L$, there exist at most $(T+1)^{Nl}$ answer sets S_a such that $\mathrm{len}(S_a) = l$. Hence, there is at least $T^{\lceil \log_T (T+1)Nl \rceil} \geq (T+1)^{Nl}$ number of prompts with a length $\lceil (\log_T(T+1)+1)Nl \rceil$. So for any answer set S_a with length $l \leq L$, we can find a prompt $\mathcal{P}_{S_a}$ with length $\lceil \log_T(T+1)Nl \rceil$ corresponding to S_a. Hence, for each $l \leq L$, we can find a different prompt $\mathcal{P}_l$ with length $\lceil \log_T L \rceil$ corresponding to such l. So, combined, for any answer set S_a with length $l \leq L$, we can find a prompt $\mathcal{P}_{\mathrm{len}(S_a)} \oplus \mathcal{P}_{S_a}$ corresponding to S_a.

Now, we will show that for any S_q satisfying the given loss α and length L, there is a transformer $\mathcal{F}$ such that $\mathcal{L}(\mathcal{F}, \mathcal{P}_{\mathrm{len}(S_a)} \oplus \mathcal{P}_{S_a} \oplus S_q, S_a) \leq \alpha$ for any S_a such that $\mathrm{len}(S_a) \leq L$. Consider that $\mathrm{len}(\mathcal{P}_{\mathrm{len}(S_a)}) + \mathrm{len}(\mathcal{P}_{S_a}) \leq (\log_T(T+1)N + \log_T L + 2)\mathrm{len}(S_a) \leq c\mathrm{len}(S_a)^d$, so this is what we want.

To prove it, we define that $(\mathcal{P}_{\mathrm{len}(S_a)} \oplus \mathcal{P}_{S_a} \oplus S_q, S_a) = \{(\mathcal{P}_{\mathrm{len}(S_a)} \oplus \mathcal{P}_{S_a} \oplus x_i, y_i)\}_{i=1}^N$, and we consider the dataset $S = \cup_{S_a:\mathrm{len}(S_a)\leq L}(\mathcal{P}_{\mathrm{len}(S_a)} \oplus \mathcal{P}_{S_a} \oplus S_q, S_a)$. It is easy to see that we only need to show that S can be memorized by a transformer with loss α.

We have that for any $(x_1, y_1), (x_2, y_2) \in S$ such that $\mathrm{len}(x_1) \geq \mathrm{len}(x_2)$, there is $x_1[1 : \mathrm{len}(x_2)] \neq x_2$. Because if $(x_1, y_1), (x_2, y_2)$ is from the same set $(\mathcal{P}_{\mathrm{len}(S_a)} \oplus \mathcal{P}_{S_a} \oplus S_q, S_a)$, then by the assumption of S_q, we know that it holds. If not, let (x_i, y_i) be from the same set $(\mathcal{P}_{\mathrm{len}(S_{a_i})} \oplus \mathcal{P}_{S_{a_i}} \oplus S_q, S_{a_i})$. If $\mathrm{len}(S_{a_1}) \neq \mathrm{len}(S_{a_2})$, since the lengths of $\mathcal{P}_{\mathrm{len}(S_{a_i})}, i \in [2]$ are the same, we know it

holds; if not, since $\mathcal{P}_{\text{len}(S_{a_1})} = \mathcal{P}_{\text{len}(S_{a_2})}$ and the lengths of $\mathcal{P}_{S_{a_1}}, i \in [2]$ are also the same, we know it still holds. We prove it.

Then by Theorem C.1 and Theorem 4.4 of (Yu et al., 2025), we know that there is a transformer $\mathcal{F}$ such that $\mathcal{F}$ can memorize S, which means that for any $(x, y) \in S$ and k, there is $\pi_{\mathcal{F}}(y[k]|x \oplus y[1 : k-1]) > \pi_{\mathcal{F}}(\sigma_j|x \oplus y[1 : k-1])$ for any $\sigma_j \neq y[k]$.

Let $\mathcal{F}_A(x) = A\mathcal{F}(x)$ where $A \in \mathbb{R}$. It is easy to see that there is a $A \in \mathbb{R}_+$ satisfied that when $c > A$, there is $-\ln \pi_{\mathcal{F}_c}(y[i]|x \oplus y[1 : i-1]) < \alpha/(L+1)$ for any $(x, y) \in S$ and i, which implies that $\mathcal{L}_{\text{nll}}(\mathcal{F}_c, x, y) \leq \alpha/(L+1)(\text{len}(S_a)+1) \leq \alpha$ for any (x, y). So $\mathcal{F}_c(c > A)$ is what we want. □

C.2 Proof of Theorem 5.2

Firstly, we have the following lemma.

Lemma C.2. *Let $x, z \in \Sigma^*$ and $\text{len}(x) = \text{len}(z)$, for any given t, w, let $x_{t,w} = t \oplus x \oplus w$ and $z_{t,w} = t \oplus z \oplus w$. Then, for any transformer $\mathcal{F}$, we have that $\lim_{\text{len}(t)\to\infty,\text{len}(w)\to\infty} ||\mathcal{F}(x_{t,w}) - \mathcal{F}(z_{t,w})||_2 = 0$.*

Proof. We assume the l-th hidden layer of $\mathcal{F}(x)$ is $\mathcal{F}_l(x)$, and the $\mathcal{F}_{l,k}(x)$ is the k-th row of $\mathcal{F}_l(x)$.

Now we assume that the following result stands for the $L-1$-th hidden layer: there is $\lim_{\text{len}(t),\text{len}(w)\to\infty} ||\mathcal{F}_{L-1,\text{len}(t)+\text{len}(x)+\text{len}(w)}(x_{t,w}) - \mathcal{F}_{L-1,\text{len}(t)+\text{len}(z)+\text{len}(w)}(z_{t,w})||_2 = 0$. Then we can show that the result is also satisfied for the L-th hidden layers.

It is easy to see that it stands for the embedding layer, which can be regarded as the 0-th hidden layer. Therefore, this result can be related to the last hidden layer, and considering that $\mathcal{F}(x_{t,w})$ only depends on the last row of the last hidden layer, this result can easily lead to the lemma.

By the definition of the transformer, we know that $\mathcal{F}_L(x_{t,w}) = \text{FNN}_L(\mathcal{F}_{L-1}(x_{t,w}) + \text{ATT}_L(\mathcal{F}_{L-1}(x_{t,w}))) + \mathcal{F}_{L-1}(x_{t,w}) + \text{ATT}_L(\mathcal{F}_{L-1}(x_{t,w}))$, similar to $\mathcal{F}_L(z_{t,w})$.

We prove this result in two parts.

The gap in the attention layer.

We will prove that $\lim_{\text{len}(t)\to\infty,\text{len}(w)\to\infty} ||ATT_L(\mathcal{F}_{L-1}(x_{t,w}))[\text{len}(x_{t,w})] - ATT_L(\mathcal{F}_{L-1}(z_{t,w}))[\text{len}(z_{t,w})]||_2 = 0$.

Firstly, we consider the attention layer, easy to see that the last row of $\text{ATT}_L(\mathcal{F}_{L-1}(x_{t,w}))$ is $\frac{\sum_{k=1}^{\text{len}(x_{t,w})} e^{u_k}\mathcal{F}_{L-1,k}(x_{t,w})}{\sum_{k=1}^{\text{len}(x_{t,w})} e^{u_k}} V_L$, and the last row of $\text{ATT}_L(\mathcal{F}_{L-1}(z_{t,w}))$ is $\frac{\sum_{k=1}^{\text{len}(z_{t,w})} e^{u'_k}\mathcal{F}_{L-1,k}(z_{t,w})}{\sum_{k=1}^{\text{len}(z_{t,w})} e^{u'_k}} V_L$. In here $u_k = \mathcal{F}_{L-1,\text{len}(x_{t,w})}(x_{t,w}) Q_L A_{\text{len}(x_{t,w})-k} K_L \mathcal{F}^T_{L-1,k}(x_{t,w})$ and $u'_k = \mathcal{F}_{L-1,\text{len}(z_{t,w})}(z_{t,w}) Q_L A_{\text{len}(z_{t,w})-k} K_L \mathcal{F}^T_{L-1,k}(z_{t,w})$.

According to the assumption and lemma, we have the following facts:

(1) Based on the assumption, there is $\lim_{\text{len}(t)\to\infty,\text{len}(w)\to\infty} ||\mathcal{F}_{L-1,\text{len}(x_{t,w})}(x_{t,w}) - \mathcal{F}_{L-1,\text{len}(z_{t,w})}(z_{t,w})||_2 \to 0$, so for any ϵ, there is a $T(\epsilon)$ such that when $\text{len}(t), \text{len}(w) \geq T(\epsilon)$, there is $||\mathcal{F}_{L-1,\text{len}(x_{t,w})}(x_{t,w}) - \mathcal{F}_{L-1,\text{len}(z_{t,w})}(z_{t,w})||_2 \leq \epsilon$.

(2) By the Lemma D.1, the $||\mathcal{F}_{L-1}(x_{t,w})||_{2,\infty}$ and $||\mathcal{F}_{L-1}(z_{t,w})||_{2,\infty}$ have the upper bound A; $||Q_L A_j K_L||_2$ also has an upper bound B.

Regarding the molecular output of the attention layer, we have that

$$\begin{aligned}
&\textstyle\sum_{k=1}^{\text{len}(x_{t,w})} e^{u_k}\mathcal{F}_{L-1,k}(x_{t,w}) \\
&= \textstyle\sum_{k=1}^{\text{len}(t)} e^{u_k}\mathcal{F}_{L-1,k}(x_{t,w}) + \sum_{k=1+\text{len}(t)}^{\text{len}(t)+\text{len}(x)} e^{u_k}\mathcal{F}_{L-1,k}(x_{t,w}) \\
&+ \textstyle\sum_{k=1+\text{len}(x)+\text{len}(t)}^{\text{len}(w)+\text{len}(x)+\text{len}(t)} e^{u_k}\mathcal{F}_{L-1,k}(x_{t,w}) \\
&\text{and} \\
&\textstyle\sum_{k=1}^{\text{len}(z_{t,w})} e^{u'_k}\mathcal{F}_{L-1,k}(z_{t,w}) \\
&= \textstyle\sum_{k=1}^{\text{len}(t)} e^{u'_k}\mathcal{F}_{L-1,k}(z_{t,w}) + \sum_{k=1+\text{len}(t)}^{\text{len}(t)+\text{len}(z)} e^{u'_k}\mathcal{F}_{L-1,k}(z_{t,w}) \\
&+ \textstyle\sum_{k=1+\text{len}(z)+\text{len}(t)}^{\text{len}(w)+\text{len}(z)+\text{len}(t)} e^{u'_k}\mathcal{F}_{L-1,k}(z_{t,w})
\end{aligned}$$

By fact (2), it holds $\sum_{k=1+\text{len}(t)}^{\text{len}(t)+\text{len}(x)} e^{u_k} \leq \text{len}(x)e^{A^2B}$ for any t, w, but

$$\lim_{\text{len}(t)\to\infty,\text{len}(w)\to\infty} \sum_{k=1}^{\text{len}(t)} e^{u_k} + \sum_{k=1+\text{len}(x)+\text{len}(t)}^{\text{len}(w)+\text{len}(x)+\text{len}(t)} e^{u_k} = \infty,$$

which is named as (*). This is similar to the $\sum_{k=1}^{\text{len}(z_{t,w})} e^{u'_k}$.

For the denominator of the attention layer output, we have a similar result; therefore, we will prove the following four results:

(r1) $\lim_{\text{len}(t)\to\infty,\text{len}(w)\to\infty} \frac{\sum_{k=1}^{\text{len}(t)} e^{u'_k}}{\sum_{k=1}^{\text{len}(t)} e^{u_k}} = 1$;

(r2) $\lim_{\text{len}(t)\to\infty,\text{len}(w)\to\infty} \frac{\sum_{k=1}^{\text{len}(t)} e^{u_k}\mathcal{F}_{L-1,k}(x_{t,w}) - \sum_{k=1}^{\text{len}(t)} e^{u'_k}\mathcal{F}_{L-1,k}(z_{t,w})}{\sum_{k=1}^{\text{len}(t)} e^{u_k}} = 0$;

(r3) $\lim_{\text{len}(t)\to\infty,\text{len}(w)\to\infty} \frac{\sum_{k=1+\text{len}(z)+\text{len}(t)}^{\text{len}(w)+\text{len}(z)+\text{len}(t)} e^{u'_k}}{\sum_{k=1+\text{len}(x)+\text{len}(t)}^{\text{len}(w)+\text{len}(x)+\text{len}(t)} e^{u_k}} = 1$;

(r4) $\lim_{\text{len}(t)\to\infty,\text{len}(w)\to\infty} \frac{\sum_{k=1+\text{len}(x)+\text{len}(t)}^{\text{len}(w)+\text{len}(x)+\text{len}(t)} e^{u_k}\mathcal{F}_{L-1,k}(x_{t,w}) - \sum_{k=1+\text{len}(z)+\text{len}(t)}^{\text{len}(w)+\text{len}(z)+\text{len}(t)} e^{u'_k}\mathcal{F}_{L-1,k}(z_{t,w})}{\sum_{k=1+\text{len}(x)+\text{len}(t)}^{\text{len}(w)+\text{len}(x)+\text{len}(t)} e^{u_k}} = 0$.

Such four results and the fact (2) can lead to what we desire, as shown below:

$$\begin{aligned}
& \lim_{\text{len}(t)\to\infty,\text{len}(w)\to\infty} ||ATT_L(\mathcal{F}_{L-1}(x_{t,w}))[\text{len}(x_{t,w})] - ATT_L(\mathcal{F}_{L-1}(z_{t,w}))[\text{len}(z_{t,w})]||_2 \\
= & \lim_{\text{len}(t)\to\infty,\text{len}(w)\to\infty} ||\frac{\sum_{k=1}^{\text{len}(x_{t,w})} e^{u_k}\mathcal{F}_{L-1,k}(x_{t,w})}{\sum_{k=1}^{\text{len}(x_{t,w})} e^{u_k}}V_L - \frac{\sum_{k=1}^{\text{len}(z_{t,w})} e^{u'_k}\mathcal{F}_{L-1,k}(z_{t,w})}{\sum_{k=1}^{\text{len}(z_{t,w})} e^{u'_k}}V_L||_2 \\
= & \lim_{\text{len}(t)\to\infty,\text{len}(w)\to\infty} ||\frac{\sum_{k=1}^{\text{len}(z_{t,w})} e^{u'_k}\mathcal{F}_{L-1,k}(z_{t,w})}{\sum_{k=1}^{\text{len}(x_{t,w})} e^{u_k}}V_L - \frac{\sum_{k=1}^{\text{len}(z_{t,w})} e^{u'_k}\mathcal{F}_{L-1,k}(z_{t,w})}{\sum_{k=1}^{\text{len}(z_{t,w})} e^{u'_k}}V_L||_2 \textbf{ (by(*), (r2), and (r4))} \\
= & \lim_{\text{len}(t)\to\infty,\text{len}(w)\to\infty} ||\frac{\sum_{k=1}^{\text{len}(z_{t,w})} e^{u'_k}\mathcal{F}_{L-1,k}(z_{t,w})}{\sum_{k=1}^{\text{len}(x_{t,w})} e^{u_k}}V_L(1 - \frac{\sum_{k=1}^{\text{len}(x_{t,w})} e^{u_k}}{\sum_{k=1}^{\text{len}(z_{t,w})} e^{u'_k}})||_2 \\
= & \ 0 \textbf{ (by (r1), (r3), and fact (2))}
\end{aligned}$$

Thus, we only need to prove such four results.

Firstly, we prove (r1) for any ϵ. By fact (1), there is a $T(\epsilon)$ such that when $\text{len}(t), \text{len}(w) \geq T(\epsilon)$, there is $||\mathcal{F}_{L-1,\text{len}(x_{t,w})}(x_{t,w}) - \mathcal{F}_{L-1,\text{len}(z_{t,w})}(z_{t,w})||_2 \leq \epsilon$. Then, we have that: $u_k - u'_k = (\mathcal{F}_{L-1,\text{len}(x_{t,w})}(x_{t,w}) - \mathcal{F}_{L-1,\text{len}(z_{t,w})}(z_{t,w}))Q_L A_{\text{len}(x_{t,w})-k)}K_L\mathcal{F}_{L-1,k}(x_{t,w})$ when $k \leq \text{len}(t)$ by lemma D.2. So when $\text{len}(t), \text{len}(w) \geq T(\epsilon)$, there is:

$$\frac{\sum_{k=1}^{\text{len}(t)} e^{u_k}}{\sum_{k=1}^{\text{len}(t)} e^{u'_k}} = \frac{\sum_{k=1}^{T(\epsilon)} e^{u_k} + \sum_{k=T(\epsilon)+1}^{\text{len}(t)} e^{u_k}}{\sum_{k=1}^{T(\epsilon)} e^{u'_k} + \sum_{k=T(\epsilon)+1}^{\text{len}(t)} e^{u'_k}} \leq \frac{T(\epsilon)e^{A^2B} + \sum_{k=T(\epsilon)+1}^{\text{len}(t)} e^{u_k}}{T(\epsilon)e^{-A^2B} + \sum_{k=T(\epsilon)+1}^{\text{len}(t)} e^{u_k}e^{-\epsilon AB}}.$$

Hence, for any $\eta > 0$, let $\epsilon \to 0$ and $\text{len}(t) >> T(\epsilon)$ satisfy $e^{-\epsilon AB} \geq 1/(1+\eta/2)$ and $\eta e^{-\epsilon AB}\sum_{k=T(\epsilon)+1}^{\text{len}(t)} e^{u_k}/2 > T(\epsilon)e^{A^2B}$. Then, there are $\frac{\sum_{k=1}^{\text{len}(t)} e^{u_k}}{\sum_{k=1}^{\text{len}(t)} e^{u'_k}} \leq \frac{T(\epsilon)e^{A^2B} + \sum_{k=T(\epsilon)+1}^{\text{len}(t)} e^{u_k}}{T(\epsilon)e^{-A^2B} + \sum_{k=T(\epsilon)+1}^{\text{len}(t)} e^{u_k}e^{-\epsilon AB}} \leq 1 + \eta$. Similarly, there is also $\frac{\sum_{k=1}^{\text{len}(t)} e^{u_k}}{\sum_{k=1}^{\text{len}(t)} e^{u'_k}} \geq \frac{T(\epsilon)e^{-A^2B} + \sum_{k=T(\epsilon)+1}^{\text{len}(t)} e^{u_k}}{T(\epsilon)e^{A^2B} + \sum_{k=T(\epsilon)+1}^{\text{len}(t)} e^{u_k}e^{\epsilon AB}} \geq 1 - \eta$ when $\epsilon \to 0$ and $\text{len}(t) >> T(\epsilon)$. Then by the arbitrariness of η, so $\lim_{\text{len}(t)\to\infty,\text{len}(w)\to\infty} \frac{\sum_{k=1}^{\text{len}(t)} e^{u_k}}{\sum_{k=1}^{\text{len}(t)} e^{u'_k}} = 1$.

Now we prove (r2):

$$\begin{aligned}
& \frac{\sum_{k=1}^{\text{len}(t)} e^{u_k}\mathcal{F}_{L-1,k}(x_{t,w}) - \sum_{k=1}^{\text{len}(t)} e^{u'_k}\mathcal{F}_{L-1,k}(z_{t,w})}{\sum_{k=1}^{\text{len}(t)} e^{u_k}} \\
\leq & \frac{\sum_{k=T(\epsilon)+1}^{\text{len}(t)} ||e^{u_k}\mathcal{F}_{L-1,k}(x_{t,w}) - e^{u'_k}\mathcal{F}_{L-1,k}(z_{t,w})||_2 + 2T(\epsilon)e^{A^2B}A}{\sum_{k=1}^{\text{len}(t)} e^{u_k}}
\end{aligned} \tag{2}$$

We have $||e^{u_k}\mathcal{F}_{L-1,k}(x_{t,w}) - e^{u'_k}\mathcal{F}_{L-1,k}(z_{t,w})||_2 \leq ||e^{u_k}\mathcal{F}_{L-1,k}(x_{t,w}) - e^{u'_k}\mathcal{F}_{L-1,k}(x_{t,w})||_2 + ||e^{u'_k}\mathcal{F}_{L-1,k}(x_{t,w}) - e^{u'_k}\mathcal{F}_{L-1,k}(z_{t,w})||_2 \leq \epsilon e^{A^2B} + (e^{\epsilon AB} - 1)Ae^{A^2B}$ when $k \geq T(\epsilon)$. Further considering that $||e^{u_k}\mathcal{F}_{L-1,k}(x_{t,w}) - e^{u'_k}\mathcal{F}_{L-1,k}(z_{t,w})||_2 / e^{u_k} \leq (\epsilon e^{A^2B} + (e^{\epsilon AB} - 1)Ae^{A^2B})/e^{u_k} = 0$ when $\epsilon \to 0$ for any k, so taking $\epsilon \to 0$ and $\text{len}(t) >> T(\epsilon)$, we can prove that $\lim_{\text{len}(t)\to\infty,\text{len}(w)\to\infty} \frac{\sum_{k=1}^{\text{len}(t)} e^{u_k}\mathcal{F}_{L-1,k}(x_{t,w}) - \sum_{k=1}^{\text{len}(t)} e^{u'_k}\mathcal{F}_{L-1,k}(z_{t,w})}{\sum_{k=1}^{\text{len}(t)} e^{u_k}} = 0$ by (2).

Now we prove (r3). for the last $\text{len}(w)$ samples, we have that $|u_k - u'_k| \leq |(\mathcal{F}_{L-1,\text{len}(x_{t,w})}(x_{t,w}) - \mathcal{F}_{L-1,\text{len}(z_{t,w})}(z_{t,w}))Q_L A_{\text{len}(x_{t,w})-k} K_L \mathcal{F}^T_{L-1,k}(x_{t,w})| + |\mathcal{F}_{L-1,\text{len}(z_{t,w})}(z_{t,w})Q_L A_{\text{len}(x_{t,w})-k} K_L (\mathcal{F}^T_{L-1,k}(x_{t,w}) - \mathcal{F}^T_{L-1,k}(z_{t,w}))|$. Thus, when $\text{len}(w), \text{len}(t) \geq T(\epsilon)$, we have that:

$$
\begin{aligned}
& \frac{\sum_{k=1+\text{len}(t)+\text{len}(x)}^{\text{len}(w)+\text{len}(t)+\text{len}(x)} e^{u_k}}{\sum_{k=1+\text{len}(t)+\text{len}(z)}^{\text{len}(w)+\text{len}(t)+\text{len}(z)} e^{u'_k}} \\
= & \frac{\sum_{k=1+\text{len}(t)+\text{len}(x)}^{T(\epsilon)+\text{len}(t)+\text{len}(x)} e^{u_k} + \sum_{k=T(\epsilon)+1+\text{len}(t)+\text{len}(x)}^{\text{len}(w)+\text{len}(t)+\text{len}(x)} e^{u_k}}{\sum_{k=1+\text{len}(t)+\text{len}(z)}^{T(\epsilon)+\text{len}(t)+\text{len}(z)} e^{u'_k} + \sum_{k=T(\epsilon)+1+\text{len}(t)+\text{len}(z)}^{\text{len}(w)+\text{len}(t)+\text{len}(z)} e^{u'_k}} \\
\leq & \frac{T(\epsilon)e^{A^2B} + \sum_{k=T(\epsilon)+1+\text{len}(t)+\text{len}(x)}^{\text{len}(w)+\text{len}(t)+\text{len}(x)} e^{u_k}}{T(\epsilon)e^{-A^2B} + \sum_{k=T(\epsilon)+1+\text{len}(t)+\text{len}(z)}^{\text{len}(w)+\text{len}(t)+\text{len}(z)} e^{u_k} e^{-2\epsilon AB}},
\end{aligned}
$$

For any $\eta > 0$, let $\epsilon \to 0$ and $\text{len}(w) >> T(\epsilon)$ such that similar to (r1), there are $\frac{\sum_{k=1+\text{len}(t)+\text{len}(x)}^{\text{len}(w)+\text{len}(t)+\text{len}(x)} e^{u_k}}{\sum_{k=1+\text{len}(t)+\text{len}(z)}^{\text{len}(w)+\text{len}(t)+\text{len}(z)} e^{u'_k}} \leq 1 + \eta$. Similarly, there is also $\frac{\sum_{k=1+\text{len}(t)+\text{len}(x)}^{\text{len}(w)+\text{len}(t)+\text{len}(x)} e^{u_k}}{\sum_{k=1+\text{len}(t)+\text{len}(z)}^{\text{len}(w)+\text{len}(t)+\text{len}(z)} e^{u'_k}} \geq 1 - \eta$, so $\lim_{\text{len}(t)\to\infty,\text{len}(w)\to\infty} \frac{\sum_{k=1+\text{len}(t)+\text{len}(x)}^{\text{len}(w)+\text{len}(t)+\text{len}(x)} e^{u_k}}{\sum_{k=1+\text{len}(t)+\text{len}(z)}^{\text{len}(w)+\text{len}(t)+\text{len}(z)} e^{u'_k}} = 1$.

For (r4), just similar to (r2). Therefore, we prove the four results.

The FNN layer.

By $\lim_{\epsilon\to 0} ||\text{FNN}(x) - \text{FNN}(x+\epsilon)||_2 = 1$, we can use the results in the attention layer and the assumptions to show that

$$
\begin{aligned}
\lim_{\text{len}(t)\to\infty,\text{len}(w)\to\infty} & ||\text{FNN}_L(\mathcal{F}_{L-1}(x_{t,w}) + \text{ATT}_L(\mathcal{F}_{L-1}(x_{t,w}))[\text{len}(x_{t,w})]) \\
& - \text{FNN}_L(\mathcal{F}_{L-1}(z_{t,w}) + \text{ATT}_L(\mathcal{F}_{L-1}(z_{t,w}))[\text{len}(z_{t,w})])||_2 = 0
\end{aligned}
$$

This is valid, because $\lim_{\text{len}(t)\to\infty,\text{len}(w)\to\infty} ||\mathcal{F}_{L-1}(z_{t,w})) - \mathcal{F}_{L-1}(x_{t,w}))||_2 = 0$ and $\lim_{\text{len}(t)\to\infty,\text{len}(w)\to\infty} ||\text{ATT}_L(\mathcal{F}_{L-1}(x_{t,w}))[\text{len}(x_{t,w})] - \text{ATT}_L(\mathcal{F}_{L-1}(z_{t,w}))[\text{len}(z_{t,w})]||_2 = 0$, as shown before.

Combining the results of such two parts is what we want. □

We now prove Theorem 5.2:

Proof. We consider the following query set: $S_q = \{x, z\} \subset \Sigma^*$, where $\text{len}(x) = \text{len}(z)$ and $x \neq z$. It is easy to see that x, z satisfies the condition in the theorem. Take $\alpha = -\ln 0.55$.

Assume there is an $\mathcal{F}$ such that for any answer set S_a, there is a $\mathcal{P}$ such that $\mathcal{L}(\mathcal{F}, \mathcal{P} \oplus S_q, S_a) \leq \alpha$. Then we consider the answer set $S_a^y = \{(y \oplus \sigma_1, y \oplus \sigma_2)\}$ where $y \in \Sigma^*$.

Firstly, for any N, we can find a y_N such that: $\mathcal{L}(\mathcal{F}, \mathcal{P} \oplus S_q, S_a^{y_N}) > \alpha$ for any $\mathcal{P}$ such that $\text{len}(\mathcal{P}) \leq N$, and $\text{len}(y_N) \geq N$. This is because for any prompt $\mathcal{P}$, there is a limited number of answer sets S_a such that $\mathcal{L}(\mathcal{F}, \mathcal{P} \oplus S_q, S_a) \leq \alpha$, and there is also a limited number of prompts with a length of at most N.

Now we consider that when $N \to \infty$, by lemma C.2, easy to see that when $\text{len}(\mathcal{P}) \geq N$ and $N \to \infty$, there is $||\mathcal{F}(\mathcal{P} \oplus x \oplus y_N) - \mathcal{F}(\mathcal{P} \oplus z \oplus y_N)||_2 \to 0$. Assume that N_0 satisfies $||\mathcal{F}(\mathcal{P} \oplus x \oplus y_{N_0}) - \mathcal{F}(\mathcal{P} \oplus z \oplus y_{N_0})||_2 < 0.05$ for any $\mathcal{P}$ such that $\text{len}(\mathcal{P}) \geq N_0$. We also have that $2\mathcal{L}(\mathcal{F}, \mathcal{P} \oplus S_q, S_a^y) \geq -\ln(\pi_{\mathcal{F}}(\sigma_1|\mathcal{P} \oplus x \oplus y_N)) - \ln(\pi_{\mathcal{F}}(\sigma_2|\mathcal{P} \oplus z \oplus y_N)) = -\ln(\pi_{\mathcal{F}}(\sigma_1|\mathcal{P} \oplus x \oplus y_N)\pi_{\mathcal{F}}(\sigma_2|\mathcal{P} \oplus z \oplus y_N))$.

Now we consider the $S_a^{y_{N_0}}$, we show that such an answer set cannot be memorized with loss α, which contradicts the assumption and leads to the theorem. By the assumption of $\mathcal{F}$, there is a $\mathcal{P}$ such that $\mathcal{L}(\mathcal{F}, \mathcal{P} \oplus S_q, S_a^{y_{N_0}}) \leq \alpha$; by the definition of y_{N_0}, there is $\text{len}(\mathcal{P}) \geq N_0$. Therefore, by the above result, there is $\pi_{\mathcal{F}}(\sigma_1|\mathcal{P} \oplus x \oplus y_{N_0}) + \pi_{\mathcal{F}}(\sigma_2|\mathcal{P} \oplus z \oplus y_{N_0}) \leq 1 + (\pi_{\mathcal{F}}(\sigma_2|\mathcal{P} \oplus z \oplus y_{N_0}) - \pi_{\mathcal{F}}(\sigma_2|\mathcal{P} \oplus x \oplus y_{N_0})) \leq 1.1$ by the assumption of N_0. Hence, using the equation $(x+y)^2/4 \geq xy$, we have that $\mathcal{L}(\mathcal{F}, \mathcal{P} \oplus S_q, S_a^{y_{N_0}}) \geq \ln(\pi_{\mathcal{F}}(\sigma_1|\mathcal{P} \oplus x \oplus y_N)\pi_{\mathcal{F}}(\sigma_2|\mathcal{P} \oplus z \oplus y_N))/2 \geq -\ln(0.55)^2/2 = -\ln 0.55$ for such a prompt $\mathcal{P}$, which contradicts the assumption; thus, we prove the theorem. □

D Proofs of Section 5.2

D.1 Proof of Theorem 5.4

We first prove several lemmas.

Lemma D.1. *For any given transformer $\mathcal{F}$, there is an $A \in \mathbb{R}_+$ that only depends on $\mathcal{F}$ such that: for any $x \in \Sigma^*$ and l, $||\mathcal{F}_l(x)||_{2,\infty} \leq A$, where $\mathcal{F}_l$ means the l-th hidden layer.*

Proof. Based on the definition of the embedding vector, for any sample x, the input of the first hidden layer is the embedding matrix v_x, whose $L_{2,\infty}$ norm has an upper bound A_0.

To prove the lemma, it suffices to show that for the l-th hidden layer, if the $L_{2,\infty}$ norm of its input has an upper bound A_{l-1} for any input x, then the $L_{2,\infty}$ norm of its output has an upper bound A_l for any input x.

Assume that $\mathcal{F}_l(x) = \text{FNN}(\mathcal{F}_{l-1}(x) + \text{ATT}(\mathcal{F}_{l-1}(x))) + \mathcal{F}_{l-1}(x) + \text{ATT}(\mathcal{F}_{l-1}(x))$.

In the attention layer, because the j-th row of the attention layer is $\frac{\sum_{i=1}^{j} e^{u_{j,i}} v_i}{\sum_{i=1}^{j} e^{u_{j,i}}} V_l$, where $u_{j,i} = v_j Q_l A_{j-i} K_l v_i^T$, v_i is the i-th row of $\mathcal{F}_{l-1}(x)$ and Q_l, K_l, V_l, A_{j-i} are parameters in the attention layer, consider that $||\frac{\sum_{i=1}^{j} e^{u_i} v_i}{\sum_{i=1}^{j} e^{u_i}}||_2 \leq \max_{i \leq j}\{||v_i||_2\}$, so $||\text{ATT}(\mathcal{F}_{l-1}(x))||_{2,\infty} \leq ||\mathcal{F}_{l-1}(x)||_{2,\infty}||V_l||_2$.

Considering that the FNN layer does not involve interaction between rows of $\mathcal{F}_{l-1}(x)$, $||\text{FNN}(x)||_{2,\infty} \leq ||W_l||_2||x||_{2,\infty} + ||b_l||_{2,\infty}$ holds for any x, where W_l and b_l are the parameters in FNN.

Combining the results of the attention layer and an FNN layer, we have that $||\mathcal{F}_l(x)||_{2,\infty} \leq ||W_l||_2(A_{l-1} + A_{l-1}||V_l||_2) + (A_{l-1} + A_{l-1}||V_l||_2) + ||b_l||_{2,\infty} = A_l$, which is what we want.

Taking $A = \max_l\{A_l\}$, we prove the lemma. □

Lemma D.2. *For any sequences x and z, if the first n symbols in x and z are the same, then for any transformer $\mathcal{F}$ and $j \in \mathbb{Z}_+$, the first n rows in the outputs of the j-th hidden layer of $\mathcal{F}(x)$ and $\mathcal{F}(z)$ are the same.*

Proof. Firstly, we prove that for the j-th hidden layer $\mathcal{F}^j$ of $\mathcal{F}$, if x_1 and z_1 satisfy that the first n rows in x_1 and z_1 are the same, then the first rows n of $\mathcal{F}^j(x_1)$ and $\mathcal{F}^j(z_1)$ are the same.

Firstly, we prove the situation for the first layer. If the first n symbols in x and z are the same, then the first n rows of their embedding matrix are also the same. Then, if it refers to the $(j-1)$-th layer, we now consider the j-th layer.

Assume that $\mathcal{F}^j$ can be written as: $\mathcal{F}^j(x) = \mathcal{F}^{j-1}(x) + \sum_{k=1}^{H} \text{softmax}(R(\mathcal{F}^{j-1}(x)Q_j, V_j\mathcal{F}^{j-1}(x)^T) + M)\mathcal{F}^{j-1}(x)K_j + \text{FNN}(\mathcal{F}^{j-1}(x) + \sum_{k=1}^{H} \text{softmax}(R(\mathcal{F}^{j-1}(x)Q_j, V_j\mathcal{F}^{j-1}(x)^T) + M)\mathcal{F}^{j-1}(x)K_j)$.

In the residual layer and FNN layer, the calculations in this layer do not include the interactions between different rows; instead, they just apply the same transformation to each row. So, when x_1 and z_1 satisfy that the first n rows in x_1 and z_1 are the same, $\text{FNN}(x_1)$ and $\text{FNN}(z_1)$ also satisfy that the first n symbols are the same.

In the attention layer, considering the definition of M, for any $i \in \mathbb{Z}_+$ and given sample x_1, we know that the i-th row of $\text{softmax}(R(xQ_k, V_k x^T) + M)$ can be written as $(x_i Q_k A_{i-1} V_k x_1^T, x_i Q_k A_{i-2} V_k x_2^T, x_i Q_k A_{i-3} V_k x_3^T, \ldots, x_i Q_k A_{i-i} V_k x_i^T, 0, 0, \ldots, 0)$, where x_i is the i-th row of x. It is easy to see that it only depends on the first i-th row of x. So when samples x and z satisfy that the first n rows in x and z are the same, then the first n rows of $\text{softmax}(R(xQ_k, V_k x^T) + M)x$ and $\text{softmax}(R(zQ_k, V_k z^T) + M)z$ are also the same. Hence, because the other parts in the attention layer do not include the interactions between different rows, we have that the whole output of the attention also satisfies that the first n symbols are the same when input x and z.

By the above result and assumption, the first n rows of

$$\sum_{k=1}^{H} \text{softmax}(R(\mathcal{F}^{j-1}(x)Q_j, V_j \mathcal{F}^{j-1}(x)^T) + M)\mathcal{F}^{j-1}(x)K_j \text{ and}$$

$$\sum_{k=1}^{H} \text{softmax}(R(\mathcal{F}^{j-1}(z)Q_j, V_j \mathcal{F}^{j-1}(z)^T) + M)\mathcal{F}^{j-1}(z)K_j$$

are the same, similar to the residual layer and the FNN layer. Combining them, we can obtain the result. Thus, we prove the lemma. □

Lemma D.3. *Let $u \in (0, 1)$ and $\mathcal{F}$ be a transformer. Then, for any $t \in \Sigma^*$ and $x, d \in \Sigma^*$, let $z \oplus t = z_t$, $x \oplus t = x_t$. we have that: $||\mathcal{F}(z_t) - \mathcal{F}(x_t)||_2 \leq \frac{C \max\{\text{len}(x), \text{len}(z)\}}{\text{len}(t)^u}$ for some constant C which depends only on $u, \mathcal{F}$.*

Proof. Let $\mathcal{F}_i$ be the i-th hidden layer of $\mathcal{F}$ and $\mathcal{F}_{i,j}$ the j-th weight of the i-th hidden layer of $\mathcal{F}$.

For convenience, let $\max\{\text{len}(x), \text{len}(z)\} = h$ in this proof. Now we assume that the following result holds for the $i-1$-th hidden layer: there exists a C_{i-1} such that $||\mathcal{F}_{i-1,\text{len}(t)+\text{len}(x)}(x_t) - \mathcal{F}_{i-1,\text{len}(t)+\text{len}(z)}(z_t)||_2 < C_{i-1}h/\text{len}(t)^u$ for any x, z, t. Then we can show that such a result is also satisfied for the i-th hidden layers.

It is easy to see that the embedding vector of x_t and z_t satisfies the above result and can be viewed as a 0-hidden layer transformer; therefore, $i = 0$ stands. Hence, consider that the output of the transformer only depends on the last row of $\mathcal{F}_L$, where L is the depth of $\mathcal{F}$. Thus, the above result can easily lead to the lemma, and we just need to prove this result.

It is easy to see that the i-th hidden layer is:

$$\mathcal{F}_i(x_t) = \text{FNN}_i(\text{ATT}_i(\mathcal{F}_{i-1}(x_t)) + \mathcal{F}_{i-1}(x_t)) + \text{ATT}_i(\mathcal{F}_{i-1}(x_t)) + \mathcal{F}_{i-1}(x_t),$$

similar to z_t. Now we will show that there is a C_i that depends only on $\mathcal{F}, u, C_{i-1}$ and satisfies $||\mathcal{F}_{i,\text{len}(t)+\text{len}(x)}(x_t) - \mathcal{F}_{i,\text{len}(t)+\text{len}(z)}(z_t)||_2 < C_i h/\text{len}(t)^u$ for any x, z, t, which is what we want. Such a result requires two parts.

The Attention Layer. Use $M[j]$ to denote the j-th row of matrix M. Regarding the attention layer, we have that:

$$\text{ATT}_i(\mathcal{F}_{i-1}(x_t))[\text{len}(x) + \text{len}(t)] = \frac{\sum_{k=1}^{\text{len}(x)+\text{len}(t)} e^{u_k} \mathcal{F}_{i-1,k}(x_t)}{\sum_{k=1}^{\text{len}(x)+\text{len}(t)} e^{u_k}} V_i,$$

and

$$\text{ATT}_i(\mathcal{F}_{i-1}(z_t))[\text{len}(z) + \text{len}(t)] = \frac{\sum_{k=1}^{\text{len}(z)+\text{len}(t)} e^{u'_k} \mathcal{F}_{i-1,k}(z_t)}{\sum_{k=1}^{\text{len}(z)+\text{len}(t)} e^{u'_k}} V_i,$$

where $u_k = R(\mathcal{F}_{i-1,\text{len}(x_t)}(x_t)Q_i, K_i \mathcal{F}^{\tau}_{i-1,k}(x_t))$ and $u'_k = R(\mathcal{F}_{i-1,\text{len}(z_t)}(z_t)Q_i, K_i\, \mathcal{F}^{\tau}_{i-1,k}(z_t))$.

By the above lemmas, we have the following facts:

(1) Let the subset transformer of $\mathcal{F}$ consist of the first $i-1$ layer named $\mathcal{F}'$. Based on this assumption, there exists a C_{i-1} such that $||\mathcal{F}_{i-1,\text{len}(x)+\text{len}(t)}(x_t) - \mathcal{F}_{i-1,\text{len}(z)+\text{len}(t)}(z_t)||_2 \leq C_{i-1}h/\text{len}(t)^u$. Hence, by Lemma D.2, we have the following fact: for any $\text{len}(t)$, there are $||\mathcal{F}_{i-1,\text{len}(x)+j}(x_t) - \mathcal{F}_{i-1,\text{len}(z)+j}(z_t)||_2 \leq C_{i-1}h/j^u$ for any j.

(2) By Lemma D.1, $||\mathcal{F}_{i-1}(x_t)||_{2,\infty}$ and $||\mathcal{F}_{i-1}(z_t)||_{2,\infty}$ have the upper bound A; $||Q_i A_j K_i||_2$ have the upper bound B for any j.

Now we consider the $||\text{ATT}_i(\mathcal{F}_{i-1}(x_t))[\text{len}(x)+\text{len}(t)] - \text{ATT}_i(\mathcal{F}_{i-1}(z_t))[\text{len}(z)+\text{len}(t)]||_2$. We will show that there is a C_i' such that $||\text{ATT}_i(\mathcal{F}_{i-1}(x_t))[\text{len}(x)+\text{len}(t)] - \text{ATT}_i(\mathcal{F}_{i-1}(z_t))[\text{len}(z)+\text{len}(t)]||_2 \leq \frac{C_i' h}{\text{len}(t)^u}$.

To be convenient, let $\sum_{k=1}^{\text{len}(x)+\text{len}(t)} e^{u_k}\mathcal{F}_{i-1,k}(x_t) = (\sum_{k=1}^{\text{len}(x)} e^{u_k}\mathcal{F}_{i-1,k}(x_t)) + (\sum_{k=\text{len}(x)+1}^{\text{len}(x)+\text{len}(t)} e^{u_k}\mathcal{F}_{i-1,k}(x_t)) = V_{1,x} + V_{2,x}$, and $\sum_{k=1}^{\text{len}(x)+\text{len}(t)} e^{u_k} = (\sum_{k=1}^{\text{len}(x)} e^{u_k}) + (\sum_{k=\text{len}(x)+1}^{\text{len}(x)+\text{len}(t)} e^{u_k}) = W_{1,x} + W_{2,x}$. Similarly, we also have $W_{1,z} + W_{2,z}$ and $V_{1,z} + V_{2,z}$.

Then we have that:

$$\begin{aligned}
& \text{ATT}_i(\mathcal{F}_{i-1}(x_t))[\text{len}(x)+\text{len}(t)] - \text{ATT}_i(\mathcal{F}_{i-1}(z_t))[\text{len}(z)+\text{len}(t)] \\
= & \frac{V_{1,x}+V_{2,x}}{W_{1,x}+W_{2,x}}V_i - \frac{V_{1,z}+V_{2,z}}{W_{1,z}+W_{2,z}}V_i \\
= & \frac{(V_{1,x}W_{1,z}-V_{1,z}W_{1,x})+(V_{1,x}W_{2,z}-V_{1,z}W_{2,x})+(V_{2,x}W_{1,z}-V_{2,z}W_{1,x})+(V_{2,x}W_{2,z}-V_{2,z}W_{2,x})}{(W_{1,z}+W_{2,z})(W_{1,x}+W_{2,x})}V_i
\end{aligned}$$

For the first part, we have the following:

$$\begin{aligned}
& ||\frac{V_{1,x}W_{1,z}-V_{1,z}W_{1,x}}{(W_{1,z}+W_{2,z})(W_{1,x}+W_{2,x})}||_2 \\
\leq & \frac{||V_{1,x}W_{1,z}||_2+||V_{1,z}W_{1,x}||_2}{(\text{len}(t)e^{-A^2B})((\text{len}(t)+h)e^{-A^2B})} \\
\leq & \frac{2W_{1,z}W_{1,x}A}{\text{len}(t)(\text{len}(t)+h)e^{-2A^2B}}.
\end{aligned}$$

By fact (2), it is easy to see that $W_{1,z}$ and $W_{1,x}$ have an upper bound $e^{A^2B}h$, so the first part $||\frac{V_{1,x}W_{1,z}-V_{1,z}W_{1,x}}{(W_{1,z}+W_{2,z})(W_{1,x}+W_{2,x})}||_2 \leq \frac{2h^2Ae^{4A^2B}}{\text{len}(t)(\text{len}(t)+h)} \leq \frac{2hAe^{4A^2B}}{\text{len}(t)}$.

For the second part,

$$\begin{aligned}
& ||\frac{V_{1,x}W_{2,z}-V_{1,z}W_{2,x}}{(W_{1,z}+W_{2,z})(W_{1,x}+W_{2,x})}||_2 \\
\leq & \frac{W_{1,x}W_{2,z}A+W_{1,z}W_{2,x}A}{(W_{1,z}+W_{2,z})(W_{1,x}+W_{2,x})} \\
= & \frac{W_{2,z}/(W_{1,z}+W_{2,z})W_{1,x}A}{W_{1,x}+W_{2,x}} + \frac{W_{2,x}/(W_{1,x}+W_{2,x})W_{1,z}A}{W_{1,z}+W_{2,z}} \\
\leq & \frac{W_{1,x}A}{W_{1,x}+W_{2,x}} + \frac{W_{1,z}A}{W_{1,z}+W_{2,z}} \\
\leq & \frac{2e^{A^2B}hA}{\text{len}(t)e^{-A^2B}}.
\end{aligned}$$

Then, similar to the first part, it is easy to see that $||\frac{V_{1,x}W_{2,z}-V_{1,z}W_{2,x}}{(W_{1,z}+W_{2,z})(W_{1,x}+W_{2,x})}||_2 \leq \frac{2hAe^{2A^2B}}{\text{len}(t)}$. It is similar to the third part.

Now we consider the last part; we have the following:

$$|\sum_{k=\text{len}(x)+1}^{\text{len}(x)+\text{len}(t)} e^{u_k} - \sum_{k=\text{len}(z)+1}^{\text{len}(z)+\text{len}(t)} e^{u_k'}| \leq \sum_{k=1}^{\text{len}(t)} |e^{u_{k+\text{len}(x)}} - e^{u_{k+\text{len}(z)}'}|.$$

Hence, by fact (1), we have that:

$$\begin{aligned}
& |u_{k+\text{len}(x)} - u_{k+\text{len}(z)}'| \\
= & |\mathcal{F}_{i-1,\text{len}(t)+\text{len}(x)}Q_iA_{\text{len}(t)-k}K_i\mathcal{F}_{i-1,k+\text{len}(x)}^T \\
& -\mathcal{F}_{i-1,\text{len}(t)+\text{len}(z)}Q_iA_{\text{len}(t)-k}K_i\mathcal{F}_{i-1,k+\text{len}(z)}| \\
\leq & |(\mathcal{F}_{i-1,\text{len}(t)+\text{len}(x)} - \mathcal{F}_{i-1,\text{len}(t)+\text{len}(z)})Q_iA_{\text{len}(t)-k}K_i\mathcal{F}_{i-1,\text{len}(x)+k}| \\
& +|\mathcal{F}_{i-1,\text{len}(t)+\text{len}(z)}Q_iA_{\text{len}(t)-k}K_i(\mathcal{F}_{i-1,\text{len}(x)+k} - \mathcal{F}_{i-1,\text{len}(z)+k})| \\
\leq & 2BA(\frac{C_{i-1}}{k^u} + \frac{C_{i-1}}{\text{len}(t)^u})h
\end{aligned}$$

So we have that: $|e^{u_{\text{len}(x)+k}} - e^{u'_{\text{len}(z)+k}}| = |e^{u_{\text{len}(x)+k}}(1 - e^{-u_{\text{len}(x)+k}+u'_{\text{len}(z)+k}})| \leq e^{A^2B}(e^{2ABh(\frac{C_{i-1}}{k^u}+\frac{C_{i-1}}{\text{len}(t)^u})} - 1)$; on the other hand, there is also $|e^{u_{\text{len}(x)+k}} - e^{u'_{\text{len}(z)+k}}| \leq e^{A^2B}$. So, combining them, we have that $|e^{u_{\text{len}(x)+k}} - e^{u'_{\text{len}(z)+k}}| \leq e^{A^2B}4ABh(\frac{C_{i-1}}{k^u} + \frac{C_{i-1}}{\text{len}(t)^u})$. So

$$\begin{aligned}
& |W_{2,x} - W_{2,z}| \\
\leq & \sum_{k=1}^{\text{len}(t)} |e^{u_{k+\text{len}(x)}} - e^{u_{k+\text{len}(z)}'}| \\
\leq & \sum_{k=1}^{\text{len}(t)} 4ABhC_{i-1}e^{A^2B}(\frac{1}{k^u} + \frac{1}{\text{len}(t)^u}) \\
\leq & 4ABhC_{i-1}e^{A^2B}(\text{len}(t)^{1-u}(1+1/(1-u))) \\
= & O(h\text{len}(t)^{1-u})
\end{aligned}$$

It is easy to see that the coefficient in O depends only on $C_{i-1}, \mathcal{F}, u$. Then, similar to before, we have that:

$$\begin{aligned} & |V_{2,x} - V_{2,z}| \\ \leq & \sum_{k=1}^{\text{len}(t)} ||(e^{u_{k+\text{len}(x)}} - e^{u'_{k+\text{len}(z)}})\mathcal{F}_{i-1,k+\text{len}(x)}(x_t)||_2 + \\ & \sum_{k=1}^{\text{len}(t)} ||e^{u'_{k+\text{len}(z)}}(\mathcal{F}_{i-1,k+\text{len}(z)}(z_t) - \mathcal{F}_{i-1,k+\text{len}(x)}(x_t))||_2 \\ \leq & \sum_{k=1}^{\text{len}(t)} |(e^{u_{k+\text{len}(x)}} - e^{u'_{k+\text{len}(z)}})|A \\ & + e^{A^2B} \sum_{k=1}^{\text{len}(t)} ||\mathcal{F}_{i-1,k+\text{len}(z)}(z_t) - \mathcal{F}_{i-1,k+\text{len}(x)}(x_t)||_2 \\ = & \sum_{k=1}^{\text{len}(t)} |(e^{u_{k+\text{len}(x)}} - e^{u'_{k+\text{len}(z)}})|A + e^{A^2B} \sum_{k=1}^{\text{len}(t)} \frac{C_{i-1}h}{k^u} \\ = & O(h\text{len}(t)^{1-u}) \end{aligned}$$

So, we write $W_\Delta = |W_{2,x} - W_{2,z}|$ and $V_\Delta = ||V_{2,x} - V_{2,z}||_2$, and we have that:

$$\begin{aligned} & ||\frac{V_{2,x}W_{2,z} - V_{2,z}W_{2,x}}{(W_{1,z}+W_{2,z})(W_{1,x}+W_{2,x})}||_2 \\ \leq & \frac{||V_{2,x}W_\Delta||_2 + |W_{2,x}V_\Delta|}{\text{len}(t)^2 e^{-2A^2B}} \\ \leq & O(\frac{h\text{len}(t)^{2-u}}{\text{len}(t)^2}) + O(h\frac{\text{len}(t)^{2-u}}{\text{len}(t)^2}) \end{aligned}$$

We use $W_{2,x} \leq \text{len}(t)e^{A^2B}$ and $||V_{2,x}||_2 \leq A\text{len}(t)e^{A^2B}$ here, and the coefficient in O depends only on A, B, C_{i-1}. So, combining such four parts, we have that $||\text{ATT}_i(\mathcal{F}_{i-1}(x_t))[\text{len}(x) + \text{len}(t)] - \text{ATT}_i(\mathcal{F}_{i-1}(z_t))[\text{len}(z) + \text{len}(t)]||_2 \leq O(\frac{h}{\text{len}(t)} + h/\text{len}(t)^u) = O(\frac{h}{\text{len}(t)^u})$, which implies that $||\text{ATT}_i(\mathcal{F}_{i-1}(x_t))[\text{len}(x) + \text{len}(t)] - \text{ATT}_i(\mathcal{F}_{i-1}(z_t))[\text{len}(z) + \text{len}(t)]||_2 \leq \frac{C'_i h}{\text{len}(t)^u}$ for some $C'_i \in \mathbb{R}_+$.

The FNN Layer.

For the FNN layer, it is easy to see that for any Relu network FNN_r with transition matrix W, there is $||\text{FNN}_r(x) - \text{FNN}_r(z)||_2 \leq ||W||_2||x - z||_2$.

So consider that $||\text{ATT}_i(\mathcal{F}_{i-1}(x_t))[\text{len}(x) + \text{len}(t)] - \text{ATT}_i(\mathcal{F}_{i-1}(z_t))[\text{len}(z) + \text{len}(t)]|| \leq C'_i h/\text{len}(t)^u$ and $||\mathcal{F}_{i,\text{len}(x)+\text{len}(t)}(x_t) - \mathcal{F}_{i,\text{len}(z)+\text{len}(t)}(z_t)||_2 \leq C_{i-1}h/\text{len}(t)^u$; it is obvious that

$$\begin{aligned} & ||\text{FNN}_i(\text{ATT}_i(\mathcal{F}_{i-1}(x_t)) + \mathcal{F}_{i-1}(x_t))[\text{len}(x) + \text{len}(t)] \\ - & \text{FNN}_i(\text{ATT}_i(\mathcal{F}_{i-1}(z_t)) + \mathcal{F}_{i-1}(z_t))[\text{len}(z) + \text{len}(t)]||_2 \\ \leq & ||W_i||_2(C'_i + C_{i-1})h/\text{len}(t)^u, \end{aligned}$$

Finally, by combining the gap from the attention layer and the FNN layer, and taking $C_i = (||W_i||_2 + 1)(C'_i + C_{i-1})$, we prove the result. □

Lemma D.4. *For any transformer $\mathcal{F}$ and $\epsilon, c \in \mathbb{R}_{>0}, d \in (0,1)$, there exists an $L \in \mathbb{Z}_{>0}$ such that if a task $S = \{(x_1, y_1)\}$ satisfies $\text{len}(y_1) \geq L$, then $|\mathcal{L}_{\text{nll}}(\mathcal{F}, x_1, y_1) - \mathcal{L}_{\text{nll}}(\mathcal{F}, \mathcal{P} \oplus x_1, y_1)| \leq \epsilon(\text{len}(y_1) + 1)$ for any $\mathcal{P}$ satisfying $\text{len}(\mathcal{P}) \leq c\text{len}(y_1)^d$.*

Proof. If not, we know that there is a ϵ such that, for any L, there is a prompt $\mathcal{P}$, query x, and answer y that satisfy $|\mathcal{L}(\mathcal{F}, \mathcal{P} \oplus x, y) - \mathcal{L}(\mathcal{F}, x, y)| > \epsilon(\text{len}(y) + 1)$, $\text{len}(y) \geq L$, and $\mathcal{P} \leq c\text{len}(y)^d$.

Let $y_j = y[1, 2, \ldots, j]$. Take a $z' \in (d, 1)$. By Lemma D.3, we know that there is a $C_\mathcal{F}$ such that $||\mathcal{F}(\mathcal{P} \oplus x \oplus y_j) - \mathcal{F}(x \oplus y_j)||_2 \leq \frac{C_\mathcal{F}\text{len}(\mathcal{P})}{(\text{len}(x)+j)^{z'}}$.

So we have that: $e^{-2\frac{C_\mathcal{F}\text{len}(\mathcal{P})}{(\text{len}(x)+j)^{z'}}} \leq \pi_\mathcal{F}(y[j+1]|\mathcal{P} \oplus x \oplus y_j)/\pi_\mathcal{F}(y[j+1]|x \oplus y_j) \leq e^{2\frac{C_\mathcal{F}\text{len}(\mathcal{P})}{(\text{len}(x)+j)^{z'}}}$; hence, $|-\ln(\pi_\mathcal{F}(y[j+1]|\mathcal{P} \oplus x \oplus y_j)) - (-\ln(\pi_\mathcal{F}(y[j+1]|x \oplus y_j)))| \leq 2\frac{C_\mathcal{F}\text{len}(\mathcal{P})}{(\text{len}(x)+j)^{z'}}$. Here, we use $y[\text{len}(y) + 1]$ to represent σ_0.

So there is:

$$\begin{aligned} & |\mathcal{L}_{\text{nll}}(\mathcal{F}, x, y) - \mathcal{L}_{\text{nll}}(\mathcal{F}, \mathcal{P} \oplus x, y)| \\ \leq & 2C_\mathcal{F}\text{len}(P) \sum_{i=0}^{\text{len}(y)} \frac{1}{(1+i)^{z'}} \\ \leq & O(\frac{\text{len}(P)}{(\text{len}(y)+1)^{z'}}(\text{len}(y) + 1)) \end{aligned}$$

By $z' > d$ and $\text{len}(\mathcal{P}) \leq c\text{len}(y)^d$, we know that for any ϵ, take $L \geq (Ac/\epsilon)^{1/(z'-d)}$ where A is the coefficient in O, there must be $\frac{\text{len}(P)}{\text{len}(y)^{z'}} \leq \epsilon/A$ for any $\text{len}(y) \leq L$, which implies that $|\mathcal{L}_{\text{nll}}(\mathcal{F}, x, y) - \mathcal{L}_{\text{nll}}(\mathcal{F}, \mathcal{P} \oplus x, y)| \leq \epsilon(\text{len}(y) + 1)$, which is contradictory to the existence of ϵ and the arbitrariness of L in the assumption, so we prove the lemma. □

We now prove Theorem 5.4.

Proof. Using Lemma D.4, we have

$$\begin{aligned} & \mathcal{L}(\mathcal{F}, S) \\ = & \tfrac{1}{|S|}\sum_{i=1}^{|S|} \mathcal{L}_{\text{nll}}(\mathcal{F}, x_i, y_i) \\ \geq & \tfrac{1}{|S|}\sum_{i=1}^{|S|} \mathcal{L}_{\text{nll}}(\mathcal{F}, \mathcal{P} \oplus x_i, y_i) - \epsilon(\text{len}(y_i) + 1) \\ \geq & \tfrac{1}{|S|}\sum_{i=1}^{|S|} \mathcal{L}_{\text{nll}}(\mathcal{F}, \mathcal{P} \oplus x_i, y_i) - \epsilon(\text{len}(S) + 1) \\ = & \mathcal{L}(\mathcal{F}, \mathcal{P} \oplus S) - \epsilon(\text{len}(S) + 1) \end{aligned}$$

which is what we want. □

D.2 Proof of Proposition 5.6 and Corollary5.12

Proof. It is easy to see that there exist at most $(T+1)^{cL^d}$ prompts with lengths no more than cL^d. For any x and prompt $\mathcal{P}$, there exist at most e^α answers y such that $\mathcal{L}_{\text{nll}}(\mathcal{F}, \mathcal{P} \oplus x, y) \leq \alpha$. There is T^L number of answers with length L. So we just need to show that $T^L > e^\alpha (T+1)^{cL^d}$.

Note that $L > (\log_T(T+1)c + \alpha)^{\frac{1}{1-d}}$ implies that $\log_T(T+1)cL^d + \alpha < L$ when $d \in (0,1)$. And $L > \frac{\alpha}{1 - c\log_T(T+1)}$ also implies that $\log_T(T+1)cL^d + \alpha < L$ when $d \in (0,1)$.

So we have that $(T+1)^{cL^d} e^\alpha \leq T^{\log_T(T+1)cL^d + \alpha} < T^L$, and Proposition 5.6 and Corollary 5.12 are proved. □

E Proofs of Section 5.3

E.1 Proofs of Theorem 5.7 and Proposition 5.8, Corollary 5.9

We need the following lemma first.

Lemma E.1. *For any transformer $\mathcal{F}$, there is a constant C such that: for any $x, z, t \in \Sigma^*$, $x_t = t \oplus x$, $z_t = t \oplus z$, $\text{len}(x) = \text{len}(z)$, and $x[\text{len}(x)] = z[\text{len}(z)]$, there is $||\mathcal{F}(x_t) - \mathcal{F}(z_t)||_2 \leq C\text{len}(x)/\text{len}(t)$.*

Proof. We assume the l-th hidden layer of $\mathcal{F}(x)$ is $\mathcal{F}_l(x)$, and the $\mathcal{F}_{l,k}(x)$ is the k-th row of $\mathcal{F}_l(x)$.

For convenience, let $h = \max\{\text{len}(x), \text{len}(z)\}$. Now we assume that the following result holds for the $L-1$-th hidden layer: there is a C_{L-1} such that $||\mathcal{F}_{L-1,\text{len}(t)+\text{len}(x)}(x_t) - \mathcal{F}_{L-1,\text{len}(t)+\text{len}(z)}(z_t)||_2 < C_{i-1}h/\text{len}(t)$ for any $x, z, t \in \Sigma^*$. Then we can show that such a result is also satisfied for the L-th hidden layers.

Consider that in the embedding layer, which can be seen as the 0-th hidden layer, by $x[\text{len}(x)] = z[\text{len}(z)]$, this result stands for the constant $C = 0$. Then consider that the output of the transformer depends only on the last row of the last attention layer; if the above result holds, it can easily lead to the lemma.

By the definition of the transformer, we know that $\mathcal{F}_L(x_t) = \text{FNN}_L(\mathcal{F}_{L-1}(x_t) + \text{ATT}_L(\mathcal{F}_{L-1}(x_t))) + \mathcal{F}_{L-1}(x_t) + \text{ATT}_L(\mathcal{F}_{L-1}(x_t))$, similar to $\mathcal{F}_L(z_t)$. We prove the result in two parts.

The attention layer gap.

Firstly, we consider the attention layer; we will show that

$$||\text{ATT}_L(\mathcal{F}_{L-1}(x_t))[\text{len}(x_t)] - \text{ATT}_L(\mathcal{F}_{L-1}(z_t))[\text{len}(z_t)]||_2 \leq C'_L h/\text{len}(t)$$

for some constant C'_L, which does not depend on x, z, t.

It is easy to see that the last row of $\text{ATT}_L(\mathcal{F}_{L-1}(x_t))$ is $\frac{\sum_{k=1}^{\text{len}(x_t)} e^{u_k}\mathcal{F}_{L-1,k}(x_t)}{\sum_{k=1}^{\text{len}(x_t)} e^{u_k}} V_L$, where $u_k = \mathcal{F}_{L-1,\text{len}(x_t)}(x_t)Q_L\ A_{\text{len}(x_t)-k}K_L\mathcal{F}_{L-1,k}(x_t)^T$.

Similarly, the last row of $\text{ATT}_L(\mathcal{F}_{L-1}(z_t))$ is $\frac{\sum_{k=1}^{\text{len}(z_t)} e^{u'_k}\mathcal{F}_{L-1,k}(z_t)}{\sum_{k=1}^{\text{len}(z_t)} e^{u'_k}} V_L$, where $u'_k = \mathcal{F}_{L-1,\text{len}(z_t)}(z_t)Q_L\ A_{\text{len}(z_t)-k}K_L\mathcal{F}_{L-1,k}(z_t)^T$.

To be convenient, let $\sum_{k=1}^{\text{len}(x_t)} e^{u_k}\mathcal{F}_{L-1,k}(x_t) = (\sum_{k=1}^{\text{len}(t)} e^{u_k}\mathcal{F}_{L-1,k}(x^m)) + (\sum_{k=\text{len}(t)+1}^{\text{len}(x_t)} e^{u_k}\mathcal{F}_{L-1,k}(x^m)) = V_{x,1} + V_{x,2}$ and $\sum_{k=1}^{\text{len}(x_t)} e^{u_k} = (\sum_{k=1}^{\text{len}(t)} e^{u_k}) + (\sum_{k=\text{len}(t)+1}^{\text{len}(x_t)} e^{u_k}) = W_{x,1} + W_{x,2}$. Similarly, we have $V_{z,i}$ and $W_{z,i}$ where $i = 1, 2$.

And by the above assumptions and lemma, we have the following facts:

(1) Based on the assumption, there is a C_{L-1} such that $||\mathcal{F}_{L-1,\text{len}(x_t)}(x_t) - \mathcal{F}_{L-1,\text{len}(z_t)}(z_t)||_2 \leq C_{L-1}h/\text{len}(t)$.

(2) By the Lemma D.1, the $||\mathcal{F}_{L-1}(x_t)||_{2,\infty}$ and $||\mathcal{F}_{L-1}(z_t)||_{2,\infty}$ have the upper bound A; $||Q_L A_j K_L||_2$ also has an upper bound B for any j.

Then we have that:

$$\begin{aligned} & \text{ATT}_L(\mathcal{F}_{L-1}(x_t))[\text{len}(x_t)] - \text{ATT}_L(\mathcal{F}_{L-1}(z_t))[\text{len}(z_t)] \\ = \ & (\tfrac{V_{x,1}+V_{x,2}}{W_{x,1}+W_{x,2}} - \tfrac{V_{z,1}+V_{z,2}}{W_{z,1}+W_{z,2}})V_L \\ = \ & \tfrac{(V_{x,1}W_{z,1}-V_{z,1}W_{x,1})+(V_{x,1}W_{z,2}-V_{z,1}W_{x,2})+(V_{x,2}W_{z,1}-V_{z,2}W_{x,1})+(V_{x,2}W_{z,2}-V_{z,2}W_{x,2})}{(W_{x,1}+W_{x,2})(W_{z,1}+W_{z,2})} V_L. \end{aligned} \tag{3}$$

Now, we will estimate the values of these four parts one by one. For the first part, let $W_\Delta = W_{z,1} - W_{x,1}$ and $V_\Delta = V_{z,1} - V_{x,1}$, we have that:

$$\begin{aligned} & ||\tfrac{V_{x,1}W_{z,1}-V_{z,1}W_{x,1}}{(W_{z,1}+W_{z,2})(W_{z,1}+W_{z,2})}||_2 \\ \leq \ & \tfrac{||V_{x,1}W_\Delta||_2+||W_{x,1}V_\Delta||_2}{(W_{z,1}+W_{z,2})(W_{z,1}+W_{z,2})} \\ \leq \ & \tfrac{|W_\Delta|A}{W_{z,1}+W_{z,2}} + \tfrac{||V_\Delta||_2}{W_{z,1}+W_{z,2}} \\ \leq \ & \tfrac{|W_\Delta|A+||V_\Delta||_2}{\text{len}(t)e^{-A^2B}}. \end{aligned} \tag{4}$$

Firstly, for $k \in [\text{len}(t)]$, we have that $|u_k - u'_k| = |(\mathcal{F}_{L-1,\text{len}(x_t)} - \mathcal{F}_{L-1,\text{len}(z_t)})\ Q_L A_{L-k} K_L \mathcal{F}_{L-1,k}^T| \leq C_{L-1}BAh/\text{len}(t)$, and consider that $u_k \leq A^2B$, by Lemma D.2 and the above facts, so there are:

$$\begin{aligned} & |W_\Delta| = |W_{x,1} - W_{z,1}| \\ \leq \ & \textstyle\sum_{k=1}^{\text{len}(t)} |e^{u_k} - e^{u'_k}| \\ = \ & \textstyle\sum_{k=1}^{\text{len}(t)} e^{u_k}|(e^{u'_k-u_k} - 1)| \\ \leq \ & 2\text{len}(t)e^{A^2B}C_{L-1}BAh/\text{len}(t) = O(h) \end{aligned}$$

Similarly, by Lemma D.2, we also have that:

$$\begin{aligned} & ||V_\Delta||_2 = ||V_{x,1} - V_{z,1}||_2 \\ \leq \ & \textstyle\sum_{i=1}^{\text{len}(t)} ||e^{u_k}\mathcal{F}_{L-1,k}(x_t) - e^{u'_k}\mathcal{F}_{L-1,k}(z_t)||_2 \\ \leq \ & \textstyle\sum_{i=1}^{\text{len}(t)} |e^{u_k} - e^{u'_k}| * ||\mathcal{F}_{L-1,k}(x_t)||_2 \\ = \ & \textstyle\sum_{i=1}^{\text{len}(t)} |e^{u_k} - e^{u'_k}|A = O(h). \end{aligned}$$

So, taking them into account in equation 4, we have $||\frac{V_{x,1}W_{z,1}-V_{z,1}W_{x,1}}{(W_{z,1}+W_{z,2})(W_{z,1}+W_{z,2})}||_2 \leq O(h/\text{len}(t))$, where the coefficient in O depends only on $\mathcal{F}$.

For part two, we have that:

$$\begin{aligned} & ||\frac{V_{x,1}W_{z,2}-V_{z,1}W_{x,2}}{(W_{z,1}+W_{z,2})(W_{z,1}+W_{z,2})}||_2 \\ \leq & \frac{||V_{x,1}W_{z,2}||_2+||W_{x,2}V_{z,1}||_2}{(W_{z,1}+W_{z,2})(W_{z,1}+W_{z,2})} \\ \leq & \frac{|W_{z,2}|A}{W_{z,1}+W_{z,2}}+\frac{||V_{z,2}||_2}{W_{z,1}+W_{z,2}} \\ \leq & \frac{2Ae^{A^2B}h}{\text{len}(t)e^{-A^2B}} = O(h/\text{len}(t)) \end{aligned} \tag{5}$$

Part three can be proved similarly. For part four, we have the following:

$$\begin{aligned} & ||\frac{V_{x,2}W_{z,2}-V_{z,2}W_{x,2}}{(W_{z,1}+W_{z,2})(W_{z,1}+W_{z,2})}||_2 \\ \leq & \frac{2(Ae^{A^2B}h)^2}{\text{len}(t)(\text{len}(t)+h)e^{-2A^2B}} \\ \leq & \frac{2(Ae^{A^2B})^2h}{\text{len}(t)e^{-2A^2B}} = O(h/\text{len}(t)) \end{aligned} \tag{6}$$

By combining such four parts into the equation 3, we have that:

$||\text{ATT}_L(\mathcal{F}_{L-1}(x_t))[\text{len}(x_t)] - \text{ATT}_L(\mathcal{F}_{L-1}(z_t))[\text{len}(z_t)]||_2 \leq O(h/\text{len}(t))$, which implies that there is a C'_L such that $||\text{ATT}_L(\mathcal{F}_{L-1}(x_t))[\text{len}(x_t)] - \text{ATT}_L(\mathcal{F}_{L-1}(z_t))[\text{len}(z_t)]||_2 \leq C'_L h/\text{len}(t)$ and C'_L only depend on $C_{L-1}, \mathcal{F}$.

The gap of the FNN layer.

Because $||\text{ATT}_L(\mathcal{F}_{L-1}(x_t))[\text{len}(x_t)] - \text{ATT}_L(\mathcal{F}_{L-1}(z_t))[\text{len}(z_t)]||_2 \leq C'_L h/\text{len}(t)$, which is so similar to the proof of theorem 5.4, it is obvious that

$$\begin{aligned} & ||\text{FNN}_L(\text{ATT}_L(\mathcal{F}_{L-1}(x_t)) + \mathcal{F}_{L-1}(x_t))[\text{len}(x)+\text{len}(t)] \\ - & \text{FNN}_L(\text{ATT}_L(\mathcal{F}_{L-1}(z_t)) + \mathcal{F}_{L-1}(z_t))[\text{len}(z)+\text{len}(t)]||_2 \\ \leq & ||W_L||_2(C'_L + C_{L-1})h/\text{len}(t), \end{aligned}$$

So, taking $C_L = (||W_L||_2 + 1)(C'_L + C_{L-1})$, we prove the result.

Now, we can prove Theorem 5.7 can be easily proved from Proposition E.2.

Finally, combining the gap from the attention layer and FNN layer, we prove the result. □

We first prove the following proposition using the lemma mentioned above.

Proposition E.2. *For any transformer $\mathcal{F}$, ϵ, there exists an (c, d) such that for any x, x', y satisfied $\text{len}(x) = \text{len}(x')$, $\text{last}(x) = \text{last}(x')$, $\text{len}(y) \geq 1$, there is $|\mathcal{L}_{nll}(\mathcal{F}, \mathcal{P}\oplus x, y) - \mathcal{L}_{nll}(\mathcal{F}, \mathcal{P}\oplus x', y)| \leq \epsilon$ for any prompt such that $\text{len}(\mathcal{P}) \geq c(\text{len}(x)+\text{len}(y))^d$.*

Proof. Firstly, for the given transformer $\mathcal{F}$, let query x, x' satisfy $\text{len}(x) = \text{len}(x')$, $\text{last}(x) = \text{last}(x')$ and answer y, prompt $\mathcal{P}$. Let $y_i = y[1] \oplus y[2] \oplus \cdots \oplus y[i]$ and $y[j]$ be the j-th symbol of y. Then we have that: $||\mathcal{F}(\mathcal{P} \oplus x \oplus y_j) - \mathcal{F}(\mathcal{P} \oplus x' \oplus y_j)|| \leq \frac{C_{\mathcal{F}}(\text{len}(x)+j)}{\text{len}(\mathcal{P})}$ according to Lemma E.1 for some constant $C_{\mathcal{F}}$; hence, there is $|-\ln(\pi_{\mathcal{F}}(y[j+1]|\mathcal{P}\oplus x\oplus y_j)) - (-\ln(\pi_{\mathcal{F}}(y[j+1]|\mathcal{P}\oplus x'\oplus y_j)))| \leq 2\frac{C_{\mathcal{F}}(\text{len}(x)+j)}{\text{len}(\mathcal{P})}$.

Let $y[\text{len}(y)+1] = \sigma_0$, so there are

$$\begin{aligned} & |\mathcal{L}_{\text{nll}}(\mathcal{F}, \mathcal{P} \oplus x, y) - \mathcal{L}_{\text{nll}}(\mathcal{F}, \mathcal{P} \oplus x', y)| \\ = & |\sum_{i=0}^{\text{len}(y)} - \ln(\pi_{\mathcal{F}}(y[j+1]|\mathcal{P} \oplus x \oplus y_j)) - (-\ln \pi_{\mathcal{F}}(y[j+1]|\mathcal{P} \oplus x' \oplus y_j))| \\ \leq & \frac{2C_{\mathcal{F}}}{\text{len}(P)} \sum_{i=0}^{\text{len}(y)} (\text{len}(x)+i) \\ \leq & O(\frac{(\text{len}(y)+1)(\text{len}(x)+\text{len}(y))}{\text{len}(P)}) \end{aligned} \tag{7}$$

When $\text{len}(P) \geq c(\text{len}(x)+\text{len}(y))^d$, there is $\frac{(\text{len}(y)+1)(\text{len}(x)+\text{len}(y))}{\text{len}(P)} \leq \frac{2(\text{len}(y)+1)^2}{\text{len}(P)} \leq \frac{2}{c(1+\text{len}(y))^{d-2}}$; so consider that $1+\text{len}(y) \geq 2$, when $d \geq 2 + \log_2^{2A_{\mathcal{F}}/c\epsilon}$, $A_{\mathcal{F}}$ is the coefficient in O which only depends on $\mathcal{F}$, there is $|\mathcal{L}_{\text{nll}}(\mathcal{F}, \mathcal{P} \oplus x, y) - \mathcal{L}_{\text{nll}}(\mathcal{F}, \mathcal{P} \oplus x', y)| \leq \epsilon$, which is what we want. So Proposition E.2 stands. □

Now, we can prove the theorem 5.7, which can be easily proved from Proposition E.2.

Proof. We just take the c, d in proposition E.2, so for any (c, d) prompt, there must be $|\mathcal{L}_{\text{nll}}(\mathcal{F}, \mathcal{P} \oplus x, y) - \mathcal{L}_{\text{nll}}(\mathcal{F}, \mathcal{P} \oplus x', y)| \leq \epsilon$, so when $\mathcal{L}_{\text{nll}}(\mathcal{F}, \mathcal{P} \oplus x, y) \leq \alpha$, there must be $\mathcal{L}_{\text{nll}}(\mathcal{F}, \mathcal{P} \oplus x', y) \leq \alpha + \epsilon$, which is what we want. □

Proposition 5.8 can be proved as follows.

Proof. Following the equation 7. When $\text{len}(P) \geq c(\text{len}(x) + \text{len}(y))^d$ and $d > 2$, there is $\frac{(\text{len}(y)+1)(\text{len}(x)+\text{len}(y))}{\text{len}(P)} \leq \frac{1}{c(\text{len}(x)+\text{len}(y))^{d-2}} \leq \epsilon$ when $\text{len}(y) \geq (1/c\epsilon)^{1/(d-2)}$, so similar to before, take $L = (A_{\mathcal{F}}/c\epsilon)^{1/(d-2)}$, the Proposition 5.8 stands. □

Corollary 5.9 can be proved as follows.

Proof. Following the equation 7, we have that:

$$\begin{aligned} & |\mathcal{L}_{\text{Nnll}}(\mathcal{F}, \mathcal{P} \oplus x, y) - \mathcal{L}_{\text{Nnll}}(\mathcal{F}, \mathcal{P} \oplus x', y)| \\ = \; & |\tfrac{1}{\text{len}(y)+1} \textstyle\sum_{i=0}^{\text{len}(y)} - \ln(\pi_{\mathcal{F}}(y[j+1]|\mathcal{P} \oplus x \oplus y_j)) - (-\ln \pi_{\mathcal{F}}(y[j+1]|\mathcal{P} \oplus x' \oplus y_j))| \\ \leq \; & \tfrac{2C_{\mathcal{F}}}{\text{len}(P)(\text{len}(y)+1)} \textstyle\sum_{i=0}^{\text{len}(y)} (\text{len}(x) + i) \\ \leq \; & O(\tfrac{\text{len}(x)+\text{len}(y)}{\text{len}(P)}) \end{aligned}$$

When $\text{len}(P) \geq c(\text{len}(x) + \text{len}(y))^d$ and $d > 1$, there is $O(\frac{\text{len}(x)+\text{len}(y)}{\text{len}(P)}) \leq O(\frac{1}{c(\text{len}(x)+\text{len}(y))^{d-1}}) \leq \epsilon$ when $\text{len}(y) \geq (A_{\mathcal{F}}/c\epsilon)^{\frac{1}{d-1}}$, where $A_{\mathcal{F}}$ in the coefficient in O. Thus, similar to before, Corollary 5.9 stands. □

E.2 Proof of Proposition 5.10

Firstly, we prove the following easy case.

Lemma E.3. *For any transformer $\mathcal{F}$, ϵ, there exists a pair of c, d such that for any task $S = \{(x_i, y_i)\}_{i=1}^N$ satisfying $\text{len}(y_i) > \text{len}(x_i)$, $\text{len}(x_i) = \text{len}(x_j)$, and $\text{last}(x_i) = \text{last}(x_j)$ for all $i, j \in [N]$, there is $|\mathcal{L}(\mathcal{F}, \mathcal{P} \oplus S) - \mathcal{L}(\mathcal{F}, \mathcal{P} \oplus S_k)| \leq \epsilon$ for any $k \in [N]$ and (c, d) prompt $\mathcal{P}$, where $S_k = \{(x_k, y_i)\}_{i=1}^N$ when $k \in [N]$.*

Proof. Lemma E.3 follows from the proof of Theorem 5.7. Let the c, d be from Theorem 5.7. We show that such L is also what we want in this corollary.

Let $S_k = \{(x_k, y_i)\}_{i=1}^N$. We have $|\mathcal{L}(\mathcal{F}, \mathcal{P} \oplus S) - \mathcal{L}(\mathcal{F}, \mathcal{P} \oplus S_k)| \leq \frac{1}{N} \sum_{i=1}^N |\mathcal{L}_{\text{nll}}(\mathcal{F}, x_i, y_i) - \mathcal{L}_{\text{nll}}(\mathcal{F}, x_k, y_i)|$. Note that $\text{len}(x_i) = \text{len}(x_k)$ and $\text{last}(x_i) = \text{last}(x_k)$. Hence, by the $\text{len}(y_i) \geq \text{len}(x_i)$, and according to the proof of Theorem 5.7, it holds $|\mathcal{L}_{\text{nll}}(\mathcal{F}, \mathcal{P} \oplus x_i, y_i) - \mathcal{L}_{\text{nll}}(\mathcal{F}, \mathcal{P} \oplus x_k, y_i)| \leq \epsilon$ for such a $\mathcal{P}$. Therefore, $|\mathcal{L}(\mathcal{F}, \mathcal{P} \oplus S) - \mathcal{L}(\mathcal{F}, \mathcal{P} \oplus S_k)| \leq \frac{1}{N} \sum_{i=1}^N \epsilon = \epsilon$. □

We now prove Proposition 5.10.

Proof. Let the c, d be from Lemma E.3. We show that such c, d is also what we want in this corollary.

Let $S = \cup_{i \in \mathbb{Z}_+} S_i$, where $S_i \subset S$ and $(x_j, y_j) \in S_i$ if and only if $\text{len}(x_j) = i$. For $S_i \neq \phi$, write $S_i' = \{(x_{i'}, y_j)\}_{(x_j, y_j) \in S_i, j \in [N]}$, where $i' = \text{argmin}_{j \in [N]}\{(x_j, y_j) \in S_i\}$.

It is easy to see that $\mathcal{L}(\mathcal{F}, S) = \frac{1}{N} \sum_{i \in \mathbb{Z}_+} \sum_{(x,y) \in S_i} \mathcal{L}_{\text{nll}}(\mathcal{F}, x, y) = \sum_{i \in \mathbb{Z}_+} \frac{|S_i| \mathcal{L}(\mathcal{F}, S_i)}{N}$.

Now we consider the i such that $S_i \neq \phi$. By Corollary E.3 and the definition of S_i', we know that when $\text{len}(y_i) > \text{len}(x_i)$ for all $i \in [N]$, it holds $|\mathcal{L}(\mathcal{F}, S_i) - \mathcal{L}(\mathcal{F}, S_i')| \leq \epsilon$. Hence, because $y_i \neq y_j$, so that $\sum_{j \in [N], (x_j, y_j) \in S_i} \pi_{\mathcal{F}}(y_j | x_i') \leq 1$, hence $\mathcal{L}(\mathcal{F}, S_i') = \frac{1}{|S_i|} \sum_{j \in [N], (x_j, y_j) \in S_i} - \ln \pi_{\mathcal{F}}(y_j | x_i') \geq \ln |S_i|$, we have that $\mathcal{L}(\mathcal{F}, S_i) \geq \ln |S_i| - \epsilon$.

Considering that $|S_i| = N_i$ and $0 \ln 0 = 0$, we have that $\mathcal{L}(\mathcal{F}, S) \geq \sum_{i \in \mathbb{Z}_+} \frac{|S_i| \mathcal{L}(\mathcal{F}, S_i)}{N} \geq \sum_{i \in \mathbb{Z}_+} \frac{N_i \ln(N_i)}{N} - \epsilon$. □

F Proofs of Section 5.4

F.1 Proof of Proposition 5.11

Proof. Without loss of generality, we consider the case that $T = 3$, that is $\Sigma = \{\sigma_i\}_{i=0}^3$. We will consider the transformer $\mathcal{F}$ with one hidden layer as follows:

(1) Embedding layer: the embedding layer of σ_i is v_i and satisfies that $v_1 = (1, 0)$ and $v_i = (0, 1)$ when $i > 1$;

(2) Hidden layer: In the attention layer, $A_j = I$, $QK = 0$, and $V = I$; the FNN layer is 0; thus, the last row of the hidden layer $\mathcal{F}_l(x)$ is calculated as $\mathcal{F}_l(x) = v_{x_{\text{len}(x)}} + \frac{\sum_{j=1}^{\text{len}(x)} v_{x_j}}{\text{len}(x)}$, where x_j is the j-th symbol in x, and v_{x_j} is the embedding vector of x_j;

(3) Output layer and FNN layer: The output layer of $\mathcal{F}$ is as follows: the weight of the second weight (corresponding to σ_2) is $\mathcal{F}_1(x) = C(\mathcal{F}_l(x)[1] - c(\mathcal{F}_l(x)[2] - 1))$, the 4-th weight (corresponding to σ_0) is $\mathcal{F}_4(x) = -C(\mathcal{F}_l(x)[1] - c(\mathcal{F}_l(x)[2] - 1))$, while others are 0. Here, C is a constant such that: $-\ln \frac{e^{-Cc}}{e^{-Cc}+e^{Cc}+2} \geq 2\alpha_2$, $\frac{2}{e^{C\frac{3\alpha_1 c}{1+c}}} \leq \alpha_1/2$; it is easy to see that such C must exist.

Let $\sigma_i^m = \sigma_i \oplus \sigma_i \oplus \sigma_i \oplus \cdots \oplus \sigma_i$. Take $x = \sigma_2^1$. Then, for any given length l, we can find a $L \geq l$ such that $\frac{[cL]-(1-4\alpha_1)cL}{[cL]+L} \geq \frac{3\alpha_1 c}{c+1}$ and $-\ln \frac{e^{-C\frac{[cL]-cL}{[cL]+L}}}{e^{C\frac{[cL]-cL}{[cL]+L}}+e^{-C\frac{[cL]-cL}{[cL]+L}}+2} \leq 2$. we consider the task $S = \{(\sigma_2^1, \sigma_2^L)\}$ and show that such a dataset is what we want. We have that:

(1) Without prompt. It is easy to see that $-\ln \pi_{\mathcal{F}}(\sigma_2|\sigma_2^K) = -\ln \frac{e^{-Cc}}{e^{-Cc}+e^{Cc}+2} \geq \alpha_2$ for any $K > 0$, so $\mathcal{L}_{\text{Nnll}}(\mathcal{F}, \sigma_2^1, \sigma_2^L) \geq \frac{\sum_{k=1}^{L} \pi_{\mathcal{F}}(\sigma_2|\sigma_2^k)}{L+1} \geq 2\alpha_2 L/(L+1) \geq \alpha_2$;

(2) With prompt. Now we consider the prompt $\sigma_1^{[cL]}$; it is easy to see that there is $-\ln \pi_{\mathcal{F}}(\sigma_2|\sigma_1^{[cL]} \oplus \sigma_2^K) = -\ln \frac{e^{C\frac{[cL]-cK}{[cL]+K}}}{e^{C\frac{[cL]-cK}{[cL]+K}}+e^{-C\frac{[cL]-cK}{[cL]+K}}+2} \leq \frac{2}{e^{C\frac{[cL]-cK}{[cL]+K}}}$ by the fact $e^{2x} > (1+x)^2$.

So when $K \leq (1-\alpha_1/4)L$, there is $-\ln \pi_{\mathcal{F}}(\sigma_2|\sigma_1^{[cL]} \oplus \sigma_2^K) \leq \frac{2}{e^{C\frac{[cL]-(1-4\alpha_1)cL}{[cL]+L}}} \leq \frac{2}{e^{C\frac{3\alpha_1 c}{c+1}}} \leq \alpha_1/2$ by the definition of L and C;

when $K \leq L$, there is $\pi_{\mathcal{F}}(\sigma_2|\sigma_1^{[cL]} \oplus \sigma_2^K) \leq -\ln \frac{e^{C\frac{[cL]-cL}{[cL]+L}}}{e^{C\frac{[cL]-cL}{[cL]+L}}+e^{-C\frac{[cL]-cL}{[cL]+L}}+2} \leq -\ln \frac{e^{-C\frac{[cL]-cL}{[cL]+L}}}{e^{C\frac{[cL]-cL}{[cL]+L}}+e^{-C\frac{[cL]-cL}{[cL]+L}}+2} \leq 2$ by the definition of L;

when $K = L$, there is also $\pi_{\mathcal{F}}(\sigma_0|\sigma_1^{[cL]} \oplus \sigma_2^L) = -\ln \frac{e^{-C\frac{[cL]-cL}{[cL]+L}}}{e^{C\frac{[cL]-cL}{[cL]+L}}+e^{-C\frac{[cL]-cL}{[cL]+L}}+2} \leq 2$ by the definition of L.

So we have

$$\begin{aligned}
&\mathcal{L}_{\text{Nnll}}(\mathcal{F}, \sigma_1^{[cL]} \oplus \sigma_2^1, \sigma_2^L) \\
&\leq \frac{\sum_{k=1}^{(1-\alpha_1/4)L} -\ln \pi_{\mathcal{F}}(\sigma_2|\sigma_1^{[cL]} \oplus \sigma_2^k)}{L+1} + \frac{\sum_{k=(1-\alpha_1/4)L+1}^{L-1} -\ln \pi_{\mathcal{F}}(\sigma_2|\sigma_1^{[cL]} \oplus \sigma_2^k) - \ln \pi_{\mathcal{F}}(\sigma_0|\sigma_1^{[cL]} \oplus \sigma_2^L)}{L+1} \\
&\leq \alpha_1/2 + \alpha_1/2 = \alpha_1.
\end{aligned}$$

Which are what we want, so we prove the result. □

F.2 Proofs of Propositions 5.13 and 5.14

We prove both propositions together.

Proof. Without loss of generality, we consider the case that $T = 3$, that is $\Sigma = \{\sigma_i\}_{i=0}^3$. Take $\alpha_1 > 0$ and $\alpha_2 > 0$ satisfied: $\alpha_1 - \ln \frac{1}{3+3/(e^{\alpha_1}-1)} < 2/3\alpha_2 - \beta$ and $0 < \alpha_1 < \alpha_2$.

We will consider the transformer $\mathcal{F}$ with one hidden layer as follows:

(1) Embedding layer: the embedding layer of σ_i is v_i and satisfies that $v_1 = (1, 0)$ and $v_i = (0, 1)$ when $i > 1$;

(2) Hidden layer: In the attention layer, $A_j = I$, $QK = 0$, and $V = I$; the FNN layer is 0; thus, the last row of the hidden layer $\mathcal{F}_l(x)$ is calculated as $\mathcal{F}_l(x) = v_{x_{\text{len}(x)}} + \frac{\sum_{j=1}^{\text{len}(x)} v_{x_j}}{\text{len}(x)}$, where x_j is the j-th symbol in x, and v_{x_j} is the embedding vector of x_j;

(3) Output layer and FNN layer: The output layer of $\mathcal{F}$ states that the weight of the second weight(corresponding to σ_2) is $\mathcal{F}_1(x) = \mathcal{F}_N(\mathcal{F}_l(x)[1])$, while others are 0. Here, $\mathcal{F}_N(x)$ is a Relu network satisfied $\mathcal{F}_N(x) = \ln(3/(e^{\alpha_1} - 1))$ when $x \geq 5/8$ and $\mathcal{F}_N(x) = \ln(3/(e^{\alpha_2} - 1))$ when $x \leq 3/5$.

We consider the situation of $c = 1$. Let $\sigma_i^m = \sigma_i \oplus \sigma_i \oplus \sigma_i \oplus \cdots \oplus \sigma_i$. Then, for any given length L, we consider the tasks $(x', y) = (\sigma_2^k, \sigma_2^{k+L})$ and $(x, y) = (\sigma_1^k, \sigma_2^{k+L})$, where $k = 2L$. We show that such a dataset and transformer are what we want; we have that:

(1) It is easy to see that for any $\mathcal{P} \subset \Sigma^*$ such that $\text{len}(\mathcal{P}) = L + k$, there is $\mathcal{F}_l(\mathcal{P} \oplus x' \oplus \sigma_2^m)[1] \leq \frac{k+L}{m+k+(k+L)} \leq \frac{1}{k/(k+L)+1} \leq 3/5$ for any $m \in \{0, 1, \ldots, k+L\}$, hence $-\ln \pi_{\mathcal{F}}(\sigma_2|\mathcal{P} \oplus x' \oplus \sigma_2^m) \geq -\ln \frac{3/(e^{\alpha_2}-1)}{3+3/(e^{\alpha_2}-1)} \geq \alpha_2$.

So $\mathcal{L}(\mathcal{F}, (\mathcal{P} \oplus x', \sigma_2^{k+L})) \geq (k+L)\alpha_2$ for any $\mathcal{P}$ with a length $L+k$; and $\mathcal{L}_{\text{Nnll}}(\mathcal{F}, (\mathcal{P} \oplus x', \sigma_2^{k+L})) \geq \frac{k+L}{k+L+1}\alpha_2 \geq 2/3\alpha_2$.

(2) Now, we take $\mathcal{P} = \sigma_1^{L+k}$; there is $\mathcal{F}_l(\mathcal{P} \oplus x \oplus \sigma_2^m)[1] \geq \frac{(k+L)+k}{m+k+(k+L)} \geq \frac{(k+L)+k}{L+k+k+(k+L)} = \frac{5}{8}$ for any $m \in \{0, 1, \ldots, k+L\}$, thus $-\ln \pi_{\mathcal{F}}(\sigma_2|\mathcal{P} \oplus x \oplus \sigma_2^m) \leq -\ln \frac{3/(e^{\alpha_1}-1)}{3+3/(e^{\alpha_1}-1)} = \alpha_1$.

So $\mathcal{L}(\mathcal{F}, (\mathcal{P} \oplus x, \sigma_2^{k+L})) \leq (k+L)\alpha_1 + \ln \pi_{\mathcal{F}}(\sigma_0|\mathcal{P} \oplus x \oplus \sigma_2^{k+L}) = (k+L)\alpha_1 - \ln \frac{1}{3+3/(e^{\alpha_1}-1)}$; and $\mathcal{L}_{\text{Nnll}}(\mathcal{F}, \mathcal{P} \oplus x, \sigma_2^{k+L}) \leq (k+L)\alpha_1/(k+L+1) - \ln \frac{1}{3+3/(e^{\alpha_1}-1)}/(k+L+1) \leq \alpha_1 - \ln \frac{1}{3+3/(e^{\alpha_1}-1)}/(k+L+1)$.

By the definition of α_1 and α_2, there is $(k+L)\alpha_1 - \ln \frac{1}{3+3/(e^{\alpha_1}-1)} < (k+L)\alpha_2 - \beta$ and $\alpha_1 - \ln \frac{1}{3+3/(e^{\alpha_1}-1)}/(K+L+1) < 2/3\alpha_2 - \beta$, which implies $\mathcal{L}_{\text{nll}}(\mathcal{F}, \mathcal{P} \oplus x', \sigma_2^{k+L}) \leq \mathcal{L}_{\text{nll}}(\mathcal{F}, \mathcal{P} \oplus x, \sigma_2^{k+L}) - \beta$ and $\mathcal{L}_{\text{Nnll}}(\mathcal{F}, \mathcal{P} \oplus x', \sigma_2^{k+L}) \leq \mathcal{L}_{\text{Nnll}}(\mathcal{F}, \mathcal{P} \oplus x, \sigma_2^{k+L}) - \beta$. We prove the result. □

G Proofs of Section 6

G.1 Proof of Theorem 6.1

We will use the following generalization bound.

Theorem G.1 (P.217 of (Mohri et al., 2018), Informal)**.** *Let the training set $\mathcal{D}_{tr}$ be iid sampled from the data distribution $\mathcal{D}_S$ and $N = |\mathcal{D}_{tr}|$. For the hypothesis space $H = \{L(\mathcal{F}(x), y) : \mathbb{R}^n \times [m] \to [0, 1]\}$ and $\delta \in \mathbb{R}_+$, with probability at least $1 - \delta$, for any $L(\mathcal{F}(x), y) \in H$, we have*

$$\mathbb{E}_{(x,y)\sim\mathcal{D}_S}[L(\mathcal{F}(x), y)] \leq \mathbb{E}_{(x,y)\in\mathcal{D}_{tr}}[L(\mathcal{F}(x), y)] + 2\text{Rad}_N^{\mathcal{D}_S}(H) + \sqrt{\frac{\ln(1/\delta)}{2N}} \tag{8}$$

Now we prove Theorem 6.1.

Proof. Let Σ_L^* be the set of $x \in \Sigma^*$ satisfying $\text{len}(x) \leq L$. For a given transformer $\mathcal{F}$, we define the hypothetical space $H_{\mathcal{F},L}$ of functions $\Sigma^* \times \Sigma \to \{0, 1\}$ as:

$$H_{\mathcal{F},L} = \{\mathcal{G}_{\mathcal{P}}(x, y) \in \Sigma^* \times \Sigma \to \{0, 1\} \mid \mathcal{G}_{\mathcal{P}}(x, y) = \mathbf{I}(\mathcal{L}_{\text{nll}}(\mathcal{F}, \mathcal{P} \oplus x, y) \geq \alpha), \mathcal{P} \in \Sigma_L^*\}.$$

It is easy to see that $H_{\mathcal{F},L}$ contains at most $(T+1)^L$ functions. Since $\mathrm{Rad}_N^{\mathcal{D}}(H) \leq \sqrt{\frac{2\ln|H|}{N}}$ (Mohri et al., 2018) for any infinite hypothetical space with range $[-1,1]$ for all the functions in it and distribution $\mathcal{D}$, the Rademacher complexity $\mathrm{Rad}_N^{\mathcal{D}}(H_{\mathcal{F},L}) \leq \sqrt{\frac{2L\ln(T+1)}{N}}$ applies to any distribution $\mathcal{D}$ of (x,y). Hence, using the Rademacher generalization bound in Theorem G.1, we prove the result. □

G.2 Proof of Theorem 6.3

Proof. Let the distribution $\mathcal{D}$ be defined as follows: $\mathcal{D}$ is a uniform distribution defined on a finite set $\mathcal{S} \subset \Sigma^* \times \Sigma^*$ satisfying $|\mathcal{S}| > \lceil N/\epsilon \rceil + 1$, $\mathrm{len}(x_1) = \mathrm{len}(x_2)$, $\mathrm{Toks}(x_1) > 1$, and $\mathrm{len}(y_1) = 1$ for all $(x_1,y_1),(x_2,y_2) \in \mathcal{S}$, where $\mathrm{Toks}(x)$ is the number of different tokens in x. Since we assumed $T \geq 3$, this is always possible due to the fact that Σ^* is infinite. For convenience, let $N/|\mathcal{S}| = \eta < \epsilon$.

Then, there are at most $C_{N/\eta}^N \leq \frac{(N/\eta)^N}{N!} \leq \frac{(N/\eta)^N}{(N/e)^N} = (e/\eta)^N$ kinds of possibilities to select subsets of $\mathcal{S}$, which contain N elements. Since we have T tokens, there exist at least $(e/\eta)^N$ prompts with a length more than $\lceil \log_T((e/\eta)^N) \rceil$. Therefore, we can find a distinct prompt $\mathcal{P}_k$ with a length of $\lceil \log_T((e/\eta)^N) \rceil = \lceil N(1-\ln\eta)/\ln T \rceil$ for each subset of $\mathcal{S}$ with N elements. If we take a $\mathcal{S}$ such that $|\mathcal{S}| = \lceil Ne/\epsilon \rceil$, then by the value of η, such length is not more than $\lceil N(2-\ln\epsilon)/\ln T \rceil$, then we can take each prompt $\mathcal{P}_k$ with any given length $L \geq N(2-\ln\epsilon)/\ln T$, which is corresponding to the conclusion in the theorem.

For $S_{w,k} \subset \mathcal{D}^N$, denote $\mathcal{P}_k$ as the corresponding prompt. Then, based on the theorem C.1 that there exists a transformer that can memorize any finite dataset (Mahdavi et al., 2023; Yu et al., 2025), there is an $\mathcal{F}$ such that for any $S_{w,k} = \{(x_{n_i}, y_{n_i})\}_{i=1}^N$, we have

(1) $\overline{\mathcal{F}}(\mathcal{P}_k \oplus x_{n_i}) = y_{n_i}$, $\overline{\mathcal{F}}(\mathcal{P}_k \oplus x_{n_i} \oplus y_{n_i}) = \sigma_0$, if $(x_{n_i}, y_{n_i}) \in S_{w,k}$;

(2) $\overline{\mathcal{F}}(\mathcal{P}_k \oplus x_{n_i}) \neq y_{n_i}$, $\overline{\mathcal{F}}(\mathcal{P}_k \oplus x_{n_i} \oplus y_{n_i}) \neq \sigma_0$, if $(x_{n_i}, y_{n_i}) \notin S_{w,k}$.

In here, $\overline{\mathcal{F}}$ is the symbol with the highest confidence in the output of $\mathcal{F}$. Let $\mathcal{F}_c$ be the transformer such that $\mathcal{F}_c(x) = c\mathcal{F}(x)$ for $c \in \mathbb{R}_{>0}$. So, by equations (1) and (2) of $\mathcal{F}$, there is an $A \in \mathbb{R}_{>0}$ such that

(c1) $\pi_{\mathcal{F}_A}(y_{n_i}|\mathcal{P}_k \oplus x_{n_i}) \geq e^{-\alpha/2}$, $\pi_{\mathcal{F}_A}(\sigma_0|\mathcal{P}_k \oplus x_{n_i} \oplus y_{n_i}) \geq e^{-\alpha/2}$, if $(x_{n_i}, y_{n_i}) \in S_{w,k}$;

(c2) $\pi_{\mathcal{F}_A}(y_{n_i}|\mathcal{P}_k \oplus x_{n_i}) < e^{-\alpha/2}$, $\pi_{\mathcal{F}_A}(\sigma_0|\mathcal{P}_k \oplus x_{n_i} \oplus y_{n_i}) < e^{-\alpha/2}$, if $(x_{n_i}, y_{n_i}) \notin S_{w,k}$.

Now we show that for any $c > A$, $\mathcal{F}_c$ and $\mathcal{D}$ are what we want.

Let $S \sim \mathcal{D}^N$ be drawn iid from $\mathcal{D}$, there must be a k such that $S \in S_{w,k}$, let $S_{w,k} = \{(x_{n_i}, y_{n_i})\}_{i=1}^N$. Then for the corresponding prompt $\mathcal{P}_k$, we have $\mathcal{L}_{\text{nll}}(\mathcal{F}, \mathcal{P}_k \oplus x_{n_j}, y_{n_j}) \leq -2\ln e^{-\alpha/2} = \alpha$ for any $(x_{n_j}, y_{n_j}) \in S_{w,k}$ by (c1). Thus,

$$\mathcal{A}_{\mathcal{F}_c,\mathcal{P}_k,S,\alpha} = \frac{1}{N}\sum_{i=1}^N \mathbf{I}(\mathcal{L}_{\text{nll}}(\mathcal{F}_c, \mathcal{P}_k \oplus x_{n_i}, y_{n_i}) \leq \alpha) = 1.$$

On the other hand, we have $\mathcal{L}_{\text{nll}}(\mathcal{F}, x_{n_j}, y_{n_j}) > -2\ln e^{-\alpha/2} = \alpha$ for any $(x_{n_j}, y_{n_j}) \notin S_{w,k}$ by (c2). Since $|S_{w,k}| = N$ and $|\mathcal{S}| \geq N/\epsilon$, we have

$$\mathcal{A}_{\mathcal{F}_c,\mathcal{P}_k,\mathcal{D},\alpha} = \mathbb{E}_{(x,y)\sim\mathcal{D}}[\mathbf{I}(\mathcal{L}_{\text{nll}}(\mathcal{F}, \mathcal{P}_k \oplus x, y) \leq \alpha)] \leq |S_{w,k}|/|\mathcal{S}| = \eta < \epsilon.$$

So $|\mathcal{A}_{\mathcal{F}_c,\mathcal{P}_k,S,\alpha} - \mathcal{A}_{\mathcal{F}_c,\mathcal{P}_k,\mathcal{D},\alpha}| \geq 1-\epsilon$, which proves the theorem. □

H Implication for Mathematical Reasoning with LLMs

Now we apply our results in section 5 to the following question: what is the capability of a prompt to obtain mathematical proofs? Some corresponding experiments are given in appendix J.

Implication 1. Short prompts such as "You are a mathematician" and "Please solve the problem step by step" have been shown to be effective in improving the reasoning ability of LLMs for some

questions. By Proposition 5.1, such prompts may effectively influence the model's output in the initial part and improve the reasoning ability of LLMs for tasks with length limitation. However, Theorem 5.4 and Corollary 5.5 show that for tasks with long proofs, if a transformer cannot solve the task, then it also cannot solve the task with high probability using such short prompts.

By Propositions 5.11, we know that a prompt with a length proportional to that of the answer may effectively produce a long proof. This suggests that we should embed the key information of the answers into the prompt. For example, we need to include the core techniques used in the proof in the prompt, and the transformer may indeed be able to piece together these techniques and combine them with the information from the query to obtain the proof.

Implication 2. By Theorem 5.7, if a transformer solves a mathematical problem using a very long prompt, then, with high probability, we have encoded the information of the answer into the prompt. However, if we want to solve many problems with different answers using one long prompt, we need to input all the information regarding the answers into the prompt at the same time. Consequently, the transformer fails to retrieve the appropriate information for each problem, which leads to a large loss value, as illustrated in Proposition 5.10.

I Loss and Pass@k Accuracy

We now explain how $L(\mathcal{F}(x), y)$ is related to Pass@k. Pass@$k(\mathcal{F}(x) = y)$ denotes the event that $\mathcal{F}(x)$ produces output y within k inference steps.

Let p denote the probability that $\mathcal{F}(x)$ produces y. Then the loss is given by $L(\mathcal{F}(x), y) = -\ln(p)$, and the probability of Pass@$k(\mathcal{F}(x) = y)$ is

$$\mathbb{P}(\text{Pass@}k(\mathcal{F}(x) = y)) = 1 - (1-p)^k = 1 - (1 - \exp^{-L(\mathcal{F}(x),y)})^k.$$

It is straightforward to observe that there is a clear monotonic relationship between the pass@k accuracy and the loss.

From the above result, we obtain that whenever the prompt $\mathcal{P}$ causes $\mathcal{L}(\mathcal{F}(\mathcal{P} \oplus x), y)$ to decrease, there must be a corresponding increase in $\mathbb{P}(\text{Pass@}k(\mathcal{F}(\mathcal{P} \oplus x) = y))$ relative to $\mathbb{P}(\text{Pass@}k(\mathcal{F}(x) = y))$. Therefore, in expectation, it achieves a higher pass@k accuracy. And the more $\mathcal{L}(\mathcal{F}(\mathcal{P} \oplus x), y)$ decreases, the more clearly we will see an improvement in $\mathbb{P}(\text{Pass@}k(\mathcal{F}(\mathcal{P} \oplus x) = y))$. Thus, in expectation, it achieves a much higher pass@k accuracy. The converse also holds.

Based on the above, we obtain that whenever $\mathcal{P}$ makes $\mathcal{L}(\mathcal{F}(\mathcal{P} \oplus x), y)$ decrease slightly, there must be a $\mathbb{P}(\text{Pass@}k(\mathcal{F}(\mathcal{P} \oplus x) = y))$ that does not grow too much relative to $\mathbb{P}(\text{Pass@}k(\mathcal{F}(x) = y))$, and conversely. Take $k = 1$ as an example. If for some prompt $\mathcal{P}$ we have $\mathcal{L}(\mathcal{F}(x), y) - \mathcal{L}(\mathcal{F}(\mathcal{P} \oplus x), y) < \delta$, then

$$\mathbb{P}(\text{Pass@}1(\mathcal{F}(\mathcal{P} \oplus x) = y)) < e^{\delta}\mathbb{P}(\text{Pass@}1(\mathcal{F}(x) = y)).$$

Thus, when δ is small (for instance, $\delta = 0.1$), $\mathbb{P}(\text{Pass@}1(\mathcal{F}(\mathcal{P} \oplus x) = y))$ can increase by at most $(e^{\delta} - 1)\mathbb{P}(\text{Pass@}1 = y)$. The same holds in the opposite direction. Therefore, if the loss function does not decrease substantially, the pass@k accuracy also cannot increase by a large amount.

Please note that we evaluate using Pass@k because it captures the model's actual success rate when multiple attempts are allowed, which is ultimately the metric that matters most to users. It also reflects how LLMs are applied in real-world scenarios, so in Section J we will perform experiments using pass@k. However, this does not stop us, or many other works, from employing cross-entropy loss for training, optimization, or analysis, since it offers a differentiable gradient signal that is tightly linked to output probabilities and accuracy, and it pushes the model to become more confident in the correct answer.

J Experiment

We emphasize that this is primarily a theory paper and that the experiments serve as proof-of-concept validation for our theoretical claims, not as large-scale empirical benchmarking.

We use 90 challenging problems from AIME-2023, AIME-2024, AIME-2025 to validate our theoretical findings and, in particular, to corroborate the observations in Appendix H. We employ the three

kinds of prompts below to solve these problems using four LLM models. The results are shown in Table 1 and 2. Two illustrative examples are given in Appendix K. For a discussion of the motivations for using pass@k accuracy and how it relates to the loss, please see Appendix I.

- **Short prompt** $\mathcal{P}_{<1}$. Input "Please provide a strict proof step by step" before each question.
- **Linear prompt** $\mathcal{P}_{=1}$. For a question x, $\mathcal{P}_{=1}$ for x is compressed from the answer of x, retaining only the core idea and removing key proof steps, resulting in a prompt with a length of about $1/10$ that of the answer.
- **Long prompt** $\mathcal{P}_{>1}$. The prompt consists of 90 linear prompts $\mathcal{P}_{=1}$ described above and is used to guide each question.

Table 1: The numbers of questions that can be solved in pass@1 using various prompts in the 90 questions. We use four models: Qwen-7B, Qwen-14B, LLaMA-70B, and GPT-OSS-120B.

Methods	Qwen-7B	Qwen-14B	LLaMA	GPT
No prompt	12/90	19/90	30/90	59/90
Prompt $\mathcal{P}_{<1}$	12/90	19/90	30/90	59/90
Prompt $\mathcal{P}_{=1}$	42/90	72/90	75/90	84/90
Prompt $\mathcal{P}_{>1}$	30/90	61/90	59/90	77/90

Table 2: The numbers of questions that can be solved in $pass@10$ using various prompts in the 90 questions. We use four models: Qwen-7B, Qwen-14B, LLaMA-70B, and GPT-OSS-120B.

Methods	Qwen-7B	Qwen-14B	LLaMA	GPT
No prompt	13/90	21/90	45/90	76/90
Prompt $\mathcal{P}_{<1}$	13/90	21/90	45/90	76/90
Prompt $\mathcal{P}_{=1}$	49/90	73/90	80/90	88/90
Prompt $\mathcal{P}_{>1}$	34/90	63/90	62/90	80/90

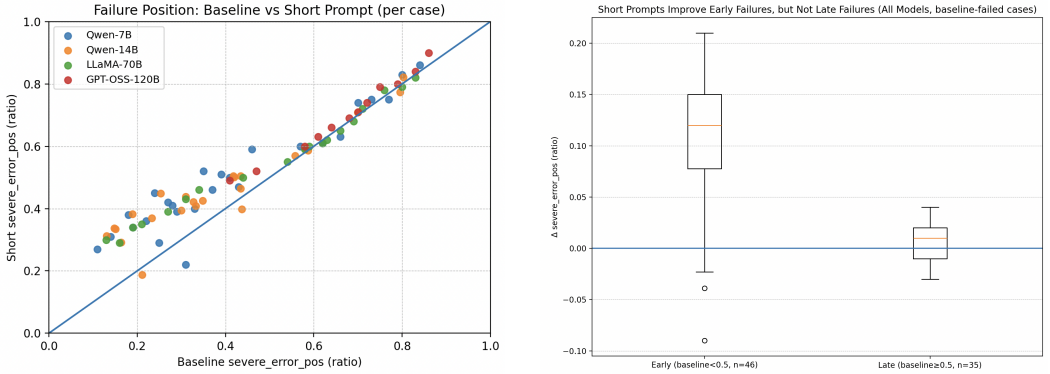


Figure 1: On the left, most of the points lie above the blue baseline, showing that in most cases, using $\mathcal{P}_{<1}$ shifts the incorrect positions back. On the right, $\mathcal{P}_{<1}$ is far more effective at correcting errors occurring at the beginning than those appearing later of the answer.

From Table 1 and 2, we draw the following conclusions that are consistent with our observations in Appendix H.

(1) Short prompts do not yield any accuracy gains compared to using no prompt at all. Through further analysis, as shown in Figure 1, short prompts indeed lead to a backward shift of the first incorrect position in the answer, verifying our observation: short prompts can influence the output in the initial part but weaken in the latter part of the answer.

(2) Comparing linear and long prompts, linear prompts achieved the best accuracy in all cases. The reason is that for the questions for which $\mathcal{P}_{=1}$ leads to correct answers but $\mathcal{P}_{>1}$ does not, the transformer with $\mathcal{P}_{>1}$ mistakenly uses proof ideas from other questions, which aligns with the observation that the "transformer fails to retrieve the appropriate information for each problem" in Appendix H.

K SPECIFIC EXAMPLES

In this section, we present two illustrative examples to demonstrate the experiment.

Compare $\mathcal{P}_{<1}$ and no prompt.

The query: Rectangles $ABCD$ and $EFGH$ are drawn such that D, E, C, F are collinear. Also, A, D, H, G all lie on a circle. If $BC = 16$, $AB = 107$,$FG = 17$, and $EF = 184$, what is the length of CE?

The answer without a prompt is given in Figure 2. The red part is the error position.

The answer with a prompt $\mathcal{P}_{<1}$ is given in Figure 3.The red part is error position.

Clearly, the short prompt $\mathcal{P}_{<1}$ makes the incorrect position shift backward.

Compare $\mathcal{P}_{=1}$ and $\mathcal{P}_{>1}$.

The query: Let x, y and z be positive real numbers that satisfy the following system of equations: $\log_2(\frac{x}{yz}) = \frac{1}{2}, \log_2(\frac{y}{xz}) = \frac{1}{3}, \log_2\left(\frac{x}{yz}\right) = \frac{1}{4}$. Then the value of $\left|\log_2(x^4y^3z^2)\right|$ is $\frac{m}{n}$ where m and n are relatively prime positive integers. Find $m + n$.

The answer without prompt $\mathcal{P}_{=1}$ is provided in Figure 4.

The prompt is: Let $a = \log 2x, b = \log 2y, c = \log 2z$. Convert to a linear system: $a - b - c = 1/2, -a + b - c = 1/3, -a - b + c = 1/4$. Solve for a,b,c; compute $|4a + 3b + 2c|$ (use add/subtract equations to isolate $2a, 2b, 2c$).

The answer for prompt $\mathcal{P}_{>1}$ is given in Figure 5

A part of the prompt $\mathcal{P}_{>1}$ is in the figure 6.

The answer follows the wrong template "split into alternating runs + stars and bars" from the $\mathcal{P}_{>1}$; subsequently, it continues to deviate, resulting in an incorrect final result. It is evident that the long prompt $\mathcal{P}_{>1}$ leads the LLM to choose the wrong problem-solving approach.

Output (Answer incorrect):
To solve for the length of \\(CE \\), we start by analyzing the given information about the rectangles \\(ABCD \\) and \\(EFGH \\).\n\nfirst, let's summarize the dimensions:\n- Rectangle \\(ABCD \\):\n - \\(BC = 16 \\)\n - \\(AB = 107 \\)\n\n- Rectangle \\(EFGH \\):\n - \\(FG = 17 \\)\n - \\(EF = 184 \\)\n\nknew know that points \\(D, E, C, F \\) are collinear, and points \\(A, D, H, G \\) lie on a circle. This implies that quadrilateral \\(ADHG \\) is cyclic.\n\since \\(D, E, C, F \\) are collinear, we can place these points on a coordinate system for easier calculation. Let's assume:\n- \\(D \\) is at the origin \\((0, 0)\\).\n- **\\(C \\) is at \\((16, 0)\\).**\n- \\(E \\) is at \\((x, 0)\\).\n- \\(F \\) is at \\((x + 17, 0)\\).\n\next, we need to determine the coordinates of \\(A \\) and \\(H \\). Since \\(A \\) is directly above \\(D \\) in rectangle \\(ABCD \\), its coordinates are:\n\\[A = (0, 107) \\]\n\similarly, since \\(H \\) is directly above \\(G \\) in rectangle \\(EFGH \\), and \\(G \\) is directly above \\(F \\), we have:\n\\[G = (x + 17, 184) \\]\n\\[H = (x + 17, 184 + 17) = (x + 17, 201) \\]\n\given that \\(A, D, H, G \\) lie on a circle, we use the property of cyclic quadrilaterals. The diagonals of a cyclic quadrilateral intersect at right angles if and only if the quadrilateral is a rectangle or a square. However, since \\(ADHG \\) is not necessarily a rectangle, we use the fact that the opposite angles sum to \\(180^\\circ \\).\n\the key insight here is to use the power of a point theorem. Specifically, the power of point \\(D \\) with respect to the circle passing through \\(A, D, H, G \\) must be equal to the power of point \\(C \\) with respect to the same circle.\n\the power of point \\(D \\) is:\n\\[DA \\cot DH = 107 \\cot 201 \\]\n\the power of point \\(C \\) is:\n\\[CA \\cot CH = 107 \\cot (201 - 16) = 107 \\cot 185 \\]\n\since \\(D \\) and \\(C \\) are collinear with \\(E \\) and \\(F \\), and the distances from \\(D \\) to \\(A \\) and \\(H \\) are known, we can set up the equation:\n\\[107 \\cot 201 = 107 \\cot 185 + CE \\cot 17 \\]\n\solving for \\(CE \\):\n\\[107 \\cot 201 = 107 \\cot 185 + CE \\cot 17 \\]\n\\[107 \\cot (201 - 185) = CE \\cot 17 \\]\n\\[107 \\cot 16 = CE \\cot 17 \\]\n\\[CE = \\frac{107 \\cot 16}{17} \\]\n\\[CE = \\frac{1712}{17} \\]\n\\[CE = 100.70588235294118 \\approx 100.71 \\]\n\nhowever, since we are dealing with exact values in geometry problems, we should keep it as a fraction:\n\\[CE = \\frac{1712}{17} \\]\n\thus, the length of \\(CE \\) is:\n\\[\n\\boxed{100}\n\\]

Figure 2: The answer without prompt. The red part is the error position.

Output (Answer incorrect):
To solve for the length of \\(CE \\), w e need to analyze the given geometric configuration and use properties of circles and rectangles.\n\nfirst, let's summarize the given information:\n- \\(ABCD \\) and \\(EFGH \\) are rectangles.\n- \\(D, E, C, F \\) are collinear.\n- \\(A, D, H, G \\) lie on a circle.\n- \\(BC = 16 \\)\n- \\(AB = 107 \\)\n- \\(FG = 17 \\)\n- \\(EF = 184 \\)\n\nsince \\(ABCD \\) and \\(EFGH \\) are rectangles, we know:\n- \\(AD = BC = 16 \\)\n- \\(CD = AB = 107 \\)\n- \\(EH = FG = 17 \\)\n- \\(GH = EF = 184 \\)\n\ngiven that \\(D, E, C, F \\) are collinear, we can place these points on a coordinate system for easier calc elation. Let's assume:\n- \\(D = (0, 0) \\)\n- \\(C = (107, 0) \\)\n- \\(E = (x, 0) \\)\n- \\(F = (x + 17, 0) \\)\n\nsince \\(A, D, H, G \\) lie on a circle, we need to find the coordinates of \\(A \\) and \\(H \\). Given \\(A = (0, 16) \\) and \\(D = (0, 0) \\), the center of the circle must be equidistant from these points. The center of the circle lies on the perpendicular bisector of \\(AD \\).\n\not he midpoint of \\(AD \\) is:\n\\[\n\\left(\\frac{0+0}{2}, \\frac{16+0}{2} \\right) = (0, 8)\n\\]\n\net the center of the circle be \\(O = (h, k) \\). Since \\(O \\) is equidistant from \\(A \\) and \\(D \\):\n\\[\no = (h, 8)\n\\]\n\the radius \\(r \\) of the circle is the distance from \\(O \\) to \\(A \\):\n\\[\nr = \\sqrt{(h-0)^2 + (8-16)^2} = \\sqrt{h^2 + 64}\n\\]\n\since \\(H \\) and \\(G \\) al so lie on this circle, we need to find their coordinates. Given \\(H = (x, 17) \\) and \\(G = (x + 17, 17) \\), the distance from \\(O \\) to \\(H \\) must equal the radius:\n\\[\n\\sqrt{(h-x)^2 + (8-17)^ 2} = \\sqrt{h^2 + 64}\n\\]\n\\[\n\\sqrt{(h-x)^2 + 81} = \\sqrt{h^2 + 64}\n\\]\n\squaring both sides:\n\\[\n(h-x)^2 + 81 = h^2 + 64\n\\]\n\\[\no^2 - 2hx + x^2 + 81 = h^2 + 64\n\\]\n\\[\n-2hx + x^2 + 81 = 64\n \\]\n\\[\nix^2 - 2hx + 17 = 0\n\\]\n\knew also know that \\(G \\) lies on the circle:\n\\[\n\\sqrt{(h-(x+17))^2 + (8-17)^2} = \\sqrt{h^2 + 64}\n\\]\n\\[\n\\sqrt{(h-x-17)^2 + 81} = \\sqrt{h^2 + 64}\n\\]\n\nanaqua ring both sides:\n\\[\n(h-x-17)^2 + 81 = h^2 + 64\n\\]\n\\[\no^2 - 34h + x^2 + 34x + 289 + 81 = h^2 + 64\n\\]\n\\[\nix^2 + 34x - 34h + 370 = 64\n\\]\n\\[\nix^2 + 34x - 34h + 306 = 0\n\\]\n\knew now have two equal tins:\n1. \\(x^2 - 2hx + 17 = 0 \\)\n2. \\(x^2 + 34x - 34h + 306 = 0 \\)\n\nsubtract the first equation from the second:\n\\[\n(x^2 + 34x - 34h + 306) - (x^2 - 2hx + 17) = 0\n\\]\n\\[\n34x - 34h + 2hx + 28 9 = 0\n\\]\n\\[\n34x + 2hx - 34h + 289 = 0\n\\]\n\\[\nix(34 + 2h) = 34h - 289\n\\]\n\\[\nix = \\frac{34h - 289}{34 + 2h}\n\\]\n\substitute \\(x \\) back into the quadratic equation:\n\\[\n\\left(\\frac{34h - 289}{34 + 2h} \\right)^2 - 2h \\left(\\frac{34h - 289}{34 + 2h} \\right) + 17 = 0\n\\]\n\solving this equation, we find:\n\\[\no = 107\n\\]\n\\[\nix = 107 - 17 = 90\n\\]\n\nthus, \\(CE = 107 - 90 = 17 \\). \n\therefore, the length of \\(CE \\) is:\n\\[\n\\boxed{17}\n\\]

Figure 3: The answer with prompt $\mathcal{P}_{<1}$. The red part is the error position.

Output (Answer correct):
Given the system of equations involving logarithms, we start by converting them into a linear system in terms of \\(a = \\log_2 x\\), \\(b = \\log_2 y\\), and \\(c = \\log_2 z\\). The given equations are:\n\n\\[\n\\log_2\\left(\\frac{x}{yz}\\right) = \\frac{1}{2}\n\\]\n\\[\n\\log_2\\left(\\frac{y}{xz}\\right) = \\frac{1}{3}\n\\]\n\\[\n\\log_2\\left(\\frac{z}{xy}\\right) = \\frac{1}{4}\n\\]\n\nUsing the properties of logarithms, these can be rewritten as:\n\n\\[\n\\log_2 x - \\log_2 y - \\log_2 z = \\frac{1}{2} \\implies a - b - c = \\frac{1}{2}\n\\]\n\\[\n\\log_2 y - \\log_2 x - \\log_2 z = \\frac{1}{3} \\implies -a + b - c = \\frac{1}{3}\n\\]\n\\[\n\\log_2 z - \\log_2 x - \\log_2 y = \\frac{1}{4} \\implies -a - b + c = \\frac{1}{4}\n\\]\n\nWe now have the following system of linear equations:\n\n\\[\na - b - c = \\frac{1}{2}\n\\]\n\\[\n-a + b - c = \\frac{1}{3}\n\\]\n\\[\n-a - b + c = \\frac{1}{4}\n\\]\n\nTo solve this system, we first add all three equations together:\n\n\\[\n(a - b - c) + (-a + b - c) + (-a - b + c) = \\frac{1}{2} + \\frac{1}{3} + \\frac{1}{4}\n\\]\n\nSimplifying the left side:\n\n\\[\na - b - c - a + b - c - a - b + c = -a - b - c\n\\]\n\nThus,\n\n\\[\n- (a + b + c) = \\frac{1}{2} + \\frac{1}{3} + \\frac{1}{4}\n\\]\n\nFinding a common denominator for the right side:\n\n\\[\n\\frac{1}{2} + \\frac{1}{3} + \\frac{1}{4} = \\frac{6}{12} + \\frac{4}{12} + \\frac{3}{12} = \\frac{13}{12}\n\\]\n\nSo,\n\n\\[\n-(a + b + c) = \\frac{13}{12} \\implies a + b + c = -\\frac{13}{12}\n\\]\n\nNext, we solve for \\(a\\), \\(b\\), and \\(c\\) individually. We use the equation \\(a + b + c = -\\frac{13}{12}\\) and substitute it into the other equations.\n\nFirst, add the first and second equations:\n\n\\[\n(a - b - c) + (-a + b - c) = \\frac{1}{2} + \\frac{1}{3}\n\\]\n\nSimplifying:\n\n\\[\n-2c = \\frac{1}{2} + \\frac{1}{3} = \\frac{5}{6} \\implies c = -\\frac{5}{12}\n\\]\n\nNext, add the first and third equations:\n\n\\[\n(a - b - c) + (-a - b + c) = \\frac{1}{2} + \\frac{1}{4}\n\\]\n\nSimplifying:\n\n\\[\n-2b = \\frac{1}{2} + \\frac{1}{4} = \\frac{3}{4} \\implies b = -\\frac{3}{8}\n\\]\n\nFinally, add the second and third equations:\n\n\\[\n(-a + b - c) + (-a - b + c) = \\frac{1}{3} + \\frac{1}{4}\n\\]\n\nSimplifying:\n\n\\[\n-2a = \\frac{1}{3} + \\frac{1}{4} = \\frac{7}{12} \\implies a = -\\frac{7}{24}\n\\]\n\nNow we have \\(a = -\\frac{7}{24}\\), \\(b = -\\frac{3}{8}\\), and \\(c = -\\frac{5}{12}\\).\n\nWe need to find \\(\\left|4a + 3b + 2c\\right|\\):\n\n\\[\n4a + 3b + 2c = 4\\left(-\\frac{7}{24}\\right) + 3\\left(-\\frac{3}{8}\\right) + 2\\left(-\\frac{5}{12}\\right)\n\\]\n\nCalculating each term:\n\n\\[\n4\\left(-\\frac{7}{24}\\right) = -\\frac{28}{24} = -\\frac{7}{6}\n\\]\n\\[\n3\\left(-\\frac{3}{8}\\right) = -\\frac{9}{8}\n\\]\n\\[\n2\\left(-\\frac{5}{12}\\right) = -\\frac{10}{12} = -\\frac{5}{6}\n\\]\n\nAdding these together:\n\n\\[\n-\\frac{7}{6} - \\frac{9}{8} - \\frac{5}{6}\n\\]\n\nFinding a common denominator (24):\n\n\\[\n-\\frac{28}{24} - \\frac{27}{24} - \\frac{20}{24} = -\\frac{75}{24} = -\\frac{25}{8}\n\\]\n\nThus,\n\n\\[\n\\left|4a + 3b + 2c\\right| = \\left|-\\frac{25}{8}\\right| = \\frac{25}{8}\n\\]\n\nThe value of \\(\\left|\\log_2(x^4y^3z^2)\\right|\\) is \\(\\frac{25}{8}\\). Therefore, \\(m = 25\\) and \\(n = 8\\), so \\(m + n = 33\\).\n\n\\[\n\\boxed{33}\n\\]

Figure 4: The correct answer with prompt $\mathcal{P}_{=1}$.

Output (Answer incorrect):
To solve this problem, we follow the "split into alternating runs + stars&bars" template from the provided context, which is commonly used when expressions exhibit structured exponent patterns.\\n\\n We are given logarithmic constraints on \\(x,y,z\\), and we are asked to compute \\(\\left|\\log_2(x^4y^3z^2)\\right|\\). At first glance this appears to be a system-of-logarithms problem, but the presence of the exponents \\(4,3,2\\) suggests an interpretation in terms of block lengths or segments.\\n\\n In particular, the expression \\(x^4y^3z^2\\) naturally decomposes into three contiguous blocks of sizes \\(4\\), \\(3\\), and \\(2\\), giving a total "path length" of \\(4+3+2=9\\). This is reminiscent of the alternating-run framework used in problems involving direction changes and compositions.\\n\\n Thus, we interpret \\(x^4y^3z^2\\) as a symbolic string consisting of \\(4\\) X-steps, \\(3\\) Y-steps, and \\(2\\) Z-steps. A standard method in such settings is to count the number of valid ways to split these symbols into alternating runs, enforcing positivity via stars and bars.\\n\\n At this point, one might attempt to linearize the logarithmic system by setting \\(a=\\log_2 x\\), etc. However, the provided context emphasizes a combinatorial run-based approach for structured exponent problems, so we proceed with that template here.\\n\\n The given system\\n\\n \\[\\log_2\\left(\\frac{x}{yz}\\right)=\\frac{1}{2},\\qquad \\log_2\\left(\\frac{y}{xz}\\right)=\\frac{1}{3},\\qquad \\log_2\\left(\\frac{z}{xy}\\right)=\\frac{1}{4}\\]\\n\\n can be loosely interpreted as fixing relative proportions between the symbols \\(X,Y,Z\\). In particular, the denominators \\(2,3,4\\) suggest an asymmetric splitting of runs, similar to the distinction between 3-run and 2-run cases in alternating-path problems (e.g., \\(RURUR\\) versus \\(URURU\\)).\\n\\n We therefore consider two symmetric alternating patterns:\\n - Pattern A: \\(XYZXYZ\\cdots\\), starting with \\(X\\);\\n - Pattern B: \\(YZX YZX\\cdots\\), starting with \\(Y\\).\\n\\n By symmetry, it suffices to count Pattern A and then multiply the result by 2.\\n\\n For Pattern A, the total number of steps of each type is fixed as follows:\\n - X-steps: \\(4\\)\\n - Y-steps: \\(3\\)\\n - Z-steps: \\(2\\)\\n\\n Splitting these into runs gives the equations\\n\\n \\[4 = x_1 + x_2 + x_3,\\qquad x_i\\ge 1\\]\\n \\[3 = y_1 + y_2,\\qquad y_i\\ge 1\\]\\n \\[2 = z_1 + z_2,\\qquad z_i\\ge 1\\]\\n\\n Using stars and bars, we compute the number of positive solutions:\\n\\n \\[\\#(x_1,x_2,x_3)=\\binom{4-1}{3-1}=\\binom{3}{2}=3\\]\\n \\[\\#(y_1,y_2)=\\binom{3-1}{2-1}=\\binom{2}{1}=2\\]\\n \\[\\#(z_1,z_2)=\\binom{2-1}{2-1}=\\binom{1}{1}=1\\]\\n\\n Thus Pattern A contributes \\(3\\cdot2\\cdot1=6\\) configurations. Accounting for Pattern B via symmetry, we obtain\\n\\n \\[2\\times6=12.\\]\\n\\n We now relate this count back to the logarithmic quantity of interest. Since \\(\\log_2(x^4y^3z^2)\\) aggregates the contributions of each block multiplicatively in base 2, it is reasonable under this template to identify its absolute value with the total number of valid run configurations.\\n\\n Therefore,\\n\\n \\[\\left|\\log_2(x^4y^3z^2)\\right| = 12.\\]\\n\\n Writing \\(12=\\frac{m}{n}\\) in lowest terms gives \\(m=12\\), \\(n=1\\), hence \\(m+n=13\\).\\n\\n (Although the original equations are logarithmic, the run-based model already enforces positivity and proportional constraints. A more detailed reconciliation with the exact fractional denominators \\(\\frac12,\\frac13,\\frac14\\) could be carried out, but this would not change the total run count under the current framework.)\\n\\n \\[\\boxed{13}\\]

Figure 5: The incorrect answer with prompt $\mathcal{P}_{>1}$.

Set up travel-time eqns: 9/s + t = 4, 9/(s+2)+t=2.4. Subtract to eliminate t: 9/s−9/(s+2)=1.6 ⇒ clear denoms ⇒ quadratic ⇒ factor for s>0, then back-sub for t. Adjust speed by +0.5 and compute 9/(s+0.5)+t. Exploit rotation on 8-cycle: condition "all blue land on previously red" implies after some rotation, rotated-blue set ⊆ original-red set ⇒ #reds ≥ #blues, so by pigeonhole \(b\le4\). Count valid colorings by \(b\): - For \(b=0,1,2,3\): argue any placement works (find a rotation sending all blues into distinct red spots). Count \(\sum_{i=0}^3 \binom{8}{i}\). - For \(b=4\): classify by adjacency pattern around the cycle and test if some rotation works. * 4 consecutive blues: works (rotate by 4); count 8. * Two adjacent pairs separated by 2: works (rotate by 2); count 4. * One adjacent pair + two isolated opposite-by-2: works (rotate by 4); count 8. * Alternating colors (0 adjacencies): works (rotate by 1); count 2. * Exclude other patterns (e.g., 3 consecutive + 1 at distance 1 or 2; two pairs distance 1; etc.). Let box sides be \(x,y,z\). Constraints: \(xyz=23\), \(xy+yz+zx=27\) (since surface area \(=54\)). Sphere diameter is space diagonal, so \[4r^2=x^2+y^2+z^2=(x+y+z)^2-2(xy+yz+zx)=(x+y+z)^2-54. \] Thus maximize \(s=x+y+z\) under the two constraints. Use symmetry/Lagrange multipliers to show at optimum at least two sides equal; WLOG set \(y=z=a\), \(x=L\). Then \[a^2L=23,\qquad a^2+2aL=27. \] Substitute \(L=23/a^2\) into second: \[a^2+\frac{46}{a}=27 \;\Rightarrow\; a^3-27a+46=0. \] Find positive roots (Rational Root Theorem + factorization); choose the smaller \(a\) to maximize \(L\) (since \(L=23/a^2\)). Then compute \(s=L+2a\) and plug into \(4r^2=s^2-54\) to get \(r^2\) (stop before final arithmetic). Use log rules to rewrite as x·log_x y=10 and 4y·log_y x=10. Multiply: 4xy(log_x y·log_y x)=100, and log_a b·log_b a=1 ⇒ 4xy=100 ⇒ solve for xy. Compute small cases to spot losing positions (P-positions) for the first player when allowed moves are {1,4}. Observe pattern repeats mod 5: losing when \(n\equiv 0,2\pmod 5\); winning otherwise. Prove via "complement to 5" strategy: if opponent takes \(a\in\{1,4\}\), respond with \(5-a\), keeping heap size invariant \(\equiv 0\) (or \(\equiv 2\)) mod 5 until terminal (0 or 2) where next player is forced. Alternatively, use recursion/Grundy: \(g(0)=0\), \(g(n)=\mathrm{mex}\{g(n-1),g(n-4)\}\), yielding period 5 with zeros at residues 0 and 2. Finally count integers \(1\le n\le 2024\) with \(n\equiv 0,2\pmod 5\): \(\lfloor 2024/5\rfloor\) for residue 0 and \(\left\lfloor(2024-2)/5\right\rfloor+1\) for residue 2, then add. Use Bayes: \(P(A\mid B)=\frac{P(B\mid A)P(A)}{P(B)}\), with \(P(B\mid A)=1\Rightarrow P(A\mid B)=\frac{P(A)}{P(B)}\). Count combinations: total \(\binom{10}{4}\). Prize event \(B\): match \(k=2,3,4\): \(\binom{4}{2}\binom{6}{2}+\binom{4}{3}\binom{6}{1}+1\). Grand prize \(A\): \(1\) way. Ratio cancels \(\binom{10}{4}\). Split the path into alternating runs: either \(RURUR\) (3 R-runs, 2 U-runs) or \(URURU\) (symmetric), so count one case and \(\times2\). For fixed pattern, choose positive run-lengths summing to 8 each: U-runs solve \(a+b=8,\ a,b\ge1\Rightarrow\) stars&bars \(\binom{7}{1}\). R-runs solve \(x+y+z=8,\ x,y,z\ge1\Rightarrow\) stars&bars \(\binom{7}{2}\). Multiply and \(\times2\). Let \(I,O\) be incenter/circumcenter; extend \(AI\) to meet \(BC\) at \(L\) and circumcircle again at \(D\). Angle chase: \(\angle BAL=\angle DAC\) and \(\angle ABL=\angle ADC\) (same arc \(AC\)) \(\Rightarrow \triangle ABL\sim\triangle ADC\), so \[\frac{AB}{AD}=\frac{AL}{AC}\ \Rightarrow\ AB\cdot AC=AL\cdot AD. \] Since \(OI\perp AD\) (center to chord), \(OI\) bisects chord \(AD\) \(\Rightarrow AD=2\,ID\), hence \[AB\cdot AC=2\,AL\cdot ID. \] Also show \(BD=CD=ID\): use \(\angle DBC=\angle DAC\), compare \(\angle IBD\) with \(\angle BID\) to get isosceles \(\triangle IBD\); symmetry for \(C\). Power of point \(L\) on circle: \(BL\cdot CL=AL\cdot LD\). Combine with similar triangles to reach \[AB\cdot AC=ID^2\cdot \frac{AL}{LD}. \] Drop perpendiculars to get right-triangle similarities giving \(DE=IF=r\) and congruence \(\triangle IGL\cong\triangle DEL\Rightarrow IL=DL\). Then \(ID=IL+DL=2IL\) and \(AL=AI+IL=ID+IL=3IL\), so \[AB\cdot AC=3\,ID^2=3\,AI^2. \] Compute \(AI^2\) via \(\triangle AOI\) right and Euler: \(OI^2=R(R-2r)\); thus \(AI^2=R^2-OI^2\). Median not in list ⇒ even length. Try n=2 fails. For n=4, mode 9 ⇒ exactly two 9s; remaining sum = 30−18=12 ⇒ other entries a,12−a, distinct. To avoid median=9, both must be <9. Integer median ⇒ a and 12−a opposite parity (larger odd). Choose valid pair, then compute sum of squares. Let a=log2x,b=log2y,c=log2z. Convert to linear system: a−b−c=1/2, −a+b−c=1/3, −a−b+c=1/4. Solve for a,b,c; compute |4a+3b+2c| (use add/subtract equations to isolate 2a,2b,2c). Use the row/column closure: if a color occupies row-set \(R\) and column-set \(C\), then every cell in \(R\times C\) must be that color (else violates "same row/col ⇒ same color"), and cells in \((R\times C^c)\cup(R^c\times C)\) must be empty; remaining cells \(R^c\times C^c\) are the other color (from "if two same-colored chips exist, their rectangle is full"). Thus mixed-color boards biject with choosing nonempty proper \(R,C\subset\{1..5\}\): count \((2^5-2)^2\). Add the 2 monochromatic full-grid cases....

Figure 6: The prompt $\mathcal{P}_{>1}$.