Sample Complexity and Representation Ability of Test-time Scaling Paradigms

Anonymous authors
Paper under double-blind review

Abstract

Test-time scaling paradigms have significantly advanced the capabilities of large language models (LLMs) on complex tasks. Despite their empirical success, theoretical understanding of the sample efficiency of various test-time strategies—such as self-consistency, best-of-n, and self-correction—remains limited. In this work, we first establish a separation result between two repeated sampling strategies: self-consistency requires $\Theta(1/\Delta^2)$ samples to produce the correct answer, while best-of-n only needs $\Theta(1/\Delta)$, where $\Delta < 1$ denotes the probability gap between the correct and second most likely answers. Next, we present an expressiveness result for the self-correction approach with verifier feedback: it enables Transformers to simulate online learning over a pool of experts at test time. Therefore, a single Transformer architecture can provably solve multiple tasks without prior knowledge of the specific task associated with a user query, extending the representation theory of Transformers from single-task to multi-task settings. Finally, we empirically validate our theoretical results, demonstrating the practical effectiveness of self-correction methods.

1 Introduction

Over the past several years, Large Language Models (LLMs) have witnessed remarkable advances, achieving unprecedented performance across a broad spectrum of application (Brown et al., 2020; Bubeck et al., 2023; Chowdhery et al., 2022). Driven by the paradigm of chain-of-thought (CoT) reasoning (Wei et al., 2022b), the outputs of LLMs have not only grown in length but also in structural complexity. In particular, recent studies have demonstrated that scaling up computational resources during test time significantly enhances the problem-solving capabilities LLMs—a phenomenon termed as the test-time scaling law (Brown et al., 2024; Wu et al., 2024; Guo et al., 2025; OpenAI, 2024b). Various methods have been proposed to effectively utilize additional test-time compute, including self-consistency (Wang et al., 2023; Brown et al., 2024; Nguyen et al., 2024; Chen et al., 2024b), best-of-n sampling (Irvine et al., 2023; Song et al., 2024a; Munkhbat et al., 2025; Qiu et al., 2024; Sessa et al., 2024), Monte Carlo Tree Search (MCTS) (Tian et al., 2024; Zhang et al., 2024d; Gao et al., 2024; Wan et al., 2024; Chen et al., 2024a; Lin et al., 2025), and self-correction (Madaan et al., 2023; Welleck et al., 2023; Chen et al., 2024d; Gou et al., 2024; Zhang et al., 2024c; Kumar et al., 2024). Powered by test-time scaling paradigms, several reasoning models, such as OpenAI-o1 (OpenAI, 2024a) and Deepseek-R1 (DeepSeek-AI, 2025), have achieved remarkable success in many complex tasks (Cobbe et al., 2021; Hendrycks et al., 2021; Shi et al., 2024; codeforce, 2025; Huang et al., 2024b; Zhang et al., 2024a).

Despite these empirical advancements, the theoretical foundations of test-time scaling remain underdeveloped. While recent progress has been made in understanding the expressiveness and learnability of chain-of-thought reasoning (Feng et al., 2023; Merrill & Sabharwal, 2023; Li et al., 2024b; Joshi et al., 2025), two fundamental challenges remain unresolved:

1. Many test-time scaling approaches rely on repeated sampling from the same LLM to select a final answer (Wang et al., 2023; Brown et al., 2024; Irvine et al., 2023; Song et al., 2024a; Nguyen et al., 2024; Chen et al., 2024b; Wu et al., 2025b; Kimi, 2025; Munkhbat et al., 2025; Qiu et al., 2024; Sessa et al., 2024). Two dominant paradigms are: self-consistency, which marginalizes reasoning paths and selects the most frequent answer; and best-of-n,

which chooses the answer with the highest reward score. However, a rigorous understanding of their sample complexities is lacking. This raises the first question:

What is the sample complexity of repeated sampling methods, particularly self-consistency and best-of-n?

2. Theoretical analyses of Transformers' expressiveness have largely focused on their ability to represent individual tasks (Yun et al., 2020; Bhattamishra et al., 2020a;b; Dehghani et al., 2018; Pérez et al., 2021; Edelman et al., 2022; Elhage et al., 2021; Likhosherstov et al., 2021; Akyürek et al., 2022; Zhao et al., 2023; Yao et al., 2021; Anil et al., 2022; Barak et al., 2022; Garg et al., 2022; Von Oswald et al., 2022; Bai et al., 2023; Olsson et al., 2022; Li et al., 2023; Garg et al., 2022; Akyürek et al., 2022; Bai et al., 2023; Von Oswald et al., 2023; Liu et al., 2022; Wei et al., 2022a; Mei & Wu, 2023; Lin et al., 2023), while the ability of Transformers to express multiple tasks at the same has been under-studied. In contrast, practical LLMs are expected to perform across diverse tasks at inference time—often using more tokens and computation than theory accounts for (Chen et al., 2024c). This gap in theory limits our understanding of test-time scaling approaches that go beyond CoT, such as self-correction (Madaan et al., 2023; Welleck et al., 2023; Chen et al., 2024d; Gou et al., 2024; Zhang et al., 2024c; Kumar et al., 2024) which uses reward information. As a result, we are motivated to pose the second central question:

How can we characterize the expressiveness under test-time scaling methods, especially in multi-task settings?

Our Contributions. This work addresses the challenges outlined above through two key contributions. First, we analyze the sample complexity of two prominent decoding strategies: self-consistency and best-of-n, in terms of the *probability gap* between the most likely (correct) and the second most likely model outputs. Our results reveal a fundamental separation in sample efficiency that highlights the advantage of the best-of-n approach.

Proposition 1.1 (Informal statement of Theorem 3.1 and Theorem 3.2)**.** *Let $\Delta \in (0, 1)$ denote the difference between the Transformer's probability of producing the correct answer and the probability of the second most likely answer. Then, self-consistency requires $\Theta(1/\Delta^2)$ samples to reliably produce the correct answer, whereas best-of-n achieves the same with only $\Theta(1/\Delta)$ samples.*

Second, we investigate Transformer's capacity for self-correction. We demonstrate that a Transformer equipped with verifier feedback at test time can implement online learning algorithms over a pool of expert models, enabling it to adaptively identify the most suitable expert and ultimately generate a response that maximizes the reward. This process is illustrated in Figure 1: given the user query (e.g. solve the PDE $\frac{1}{c(x)^2}\frac{\partial^2 u}{\partial t^2} - \Delta u = 0$ in $\Omega \times (0, T)$ with some boundary conditions), the Transformer f autoregressively generates a sequence of actions (e.g., selecting the sixth expert) and responses (e.g., constructing and applying a spectral method solver), conditioned on the history of previous action-response pairs and their corresponding rewards (e.g., solution error). Notably, this process relies solely on the Transformer f—whose architecture encapsulates the capabilities of all experts—and the reward function, distinguishing it from traditional routing algorithms that explicitly query experts. As such, this mechanism allows a single Transformer architecture to solve multiple tasks without prior knowledge of the specific task associated with a user query.

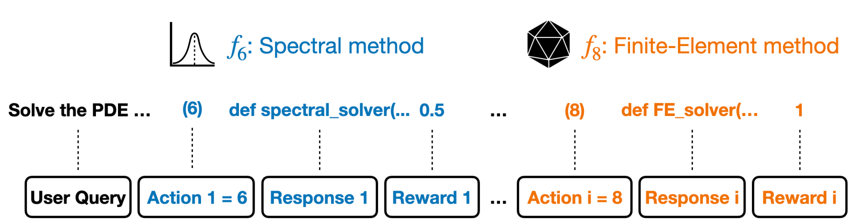

Figure 1: An illustration of test-time online learning (figure adapted from (Li et al., 2025)), where the Transformer progressively learns that finite-element method solves the partial differential equation with higher accuracy.

Proposition 1.2 (Informal statement of Theorem 4.7)**.** *There exists a generic way to construct a wider transformer f from any Transformer-based expert models $f_1, \dots, f_E$ such that, when provided with reward-based feedback, f can generate a sequence of responses where the t-th response has regret $o(1)$.*

Proposition 1.2 has two key implications. First, it demonstrates that a Transformer can express multiple tasks within a single architecture, extending beyond prior theoretical results that focus on single-task expressiveness. Importantly, the construction is task-agnostic and independent of the specific expert Transformers used, making both the result and the underlying techniques of independent theoretical interest. Second, Proposition 1.2 reveals a fundamental distinction between self-correction and repeated-sampling paradigms. While repeated-sampling methods generate identically distributed responses across attempts, self-correction *provably* allows the model to update its attempts based on verifier feedback, thereby increasing the probability of producing the correct answer as inference progresses. We further validate this results through controlled experiments.

2 Preliminaries

Transformers. In this work, we consider attention-only Transformers defined as follows.

Definition 2.1 (Transformer)**.** We define a Transformer model over vocabulary $\mathcal{V}$ as a tuple

$$(\theta, \mathrm{pe}, (\mathbf{K}_h^{(l)}, \mathbf{Q}_h^{(l)}, \mathbf{V}_h^{(l)})_{h\in[H], l\in[L]}, \vartheta, \mathcal{V})$$

where $\theta : \mathcal{V} \to \mathbb{R}^d$ is the tokenizer, $\mathrm{pe} : \mathbb{R}^d \times \mathcal{V}^\omega \to \mathbb{R}^d$ is a position encoder, $\mathbf{K}_h^{(l)}, \mathbf{Q}_h^{(l)}, \mathbf{V}_h^{(l)} \in \mathbb{R}^{d\times d}$ are the key, query, value matrices over L layers and H heads each layer, and ϑ is the output feature. The computation of a Transformer rolls out as follows:

1. For each $i = 1, \dots, n$, $X_i^{(1)} = \mathrm{pe}(\theta(v_i); v_1, \dots, v_i)$.
2. For each $l = 1, \dots, L$, compute each $X_i^{(l+1)}$ for $i = 1, \dots, n$ by
$$X_i^{(l+1)} = \sum_{h=1}^{H} \sum_{j=1}^{i} \frac{\exp\left(s_h^{(l)}(X_i, X_j)\right)}{Z_h^{(l)}} \cdot \mathbf{V}_h^{(l)} X_j^{(l)}, \tag{1}$$
where $s_h^{(l)}(\cdot)$ is the attention score defined by $s_h^{(l)}(X_i, X_j) = (\mathbf{Q}_h^{(l)} X_i^{(l)})^\top (\mathbf{K}_h^{(l)} X_j^{(l)})$ and $Z_h^{(l)} = \sum_{j=1}^{i} \exp\left(s_h^{(l)}(X_i, X_j)\right)$ is the normalizing constant.
3. The output probability is given by
$$p_f(y|v_1, \dots, v_n) = \mathrm{Softmax}(\vartheta(y)^\top X_n^{(L)}),\ y \in \mathcal{V}.$$

In particular, we assume the softmax attention layer has precision ϵ: if two attention scores s_1, s_2 satisfy $e^{s_1} < \epsilon \cdot e^{s_2}$, then e^{s_1} is treated as zero in the attention computation of Eq. (1).

While classical positional encoders is solely dependent on the index of the current token (i.e. we may write $\mathrm{pe}(\theta(v_i); v_1, \dots, v_i) = \mathrm{pe}(\theta(v_i); i)$), recent advance (He et al., 2024; Zhang et al., 2024b; Golovneva et al., 2024) has extended this notion to incorporate set membership information of preceding tokens. This generalization proves crucial for enhancing the long-context capability required for effective self-correction. Motivated by this insight, we introduce the following notion of a generalized position encoder.

Definition 2.2 (Generalized Position Encoder)**.** We say that $\mathrm{pe} : \mathbb{R}^d \times \mathcal{V}^\omega \to \mathbb{R}^d$ is a generalized position encoder w.r.t. a partition $\mathcal{V}_1, \dots, \mathcal{V}_K$ of $\mathcal{V}$ if it maps an input feature in $\mathbb{R}^d$ and a token sequence (of arbitrary length) $v_1, \cdots, v_i$ to a vector in $\mathbb{R}^d$, so that it only depends on the input feature and the membership of each v_i in the sets $\mathcal{V}_1, \dots, \mathcal{V}_K$, i.e.

$$\mathrm{pe}(\theta(v_i); v_1, \dots, v_i) = \mathrm{pe}\left(\theta(v_i); (\mathbb{1}(v_j \in \mathcal{V}_k))_{j\in[i], k\in[K]}\right).$$

Test-time scaling. In this work, we study the following three strategies for test-time scaling.

1. *Self-consistency* samples n i.i.d. responses from the language model and chooses the most consistent answer, while marginalizing over the reasoning paths.
2. *Best-of-n* samples n i.i.d. responses from the language model and chooses the answer with the highest score given by the reward model.
3. In the *Self-Correction* paradigm, the Transformer autonomously generates a sequence of n responses, each conditioned on the previous responses and their respective reward scores.

3 Separation between Self-Consistency and Best-of-n

In this section, we study the sample complexity of self-consistency and best-of-n. Let q denote the user query (e.g. a math problem) and $\mathcal{O}$ denote the answer space; then for each answer $o \in \mathcal{O}$ we define $p(o)$ as the marginalized probability of generating o over all possible reasoning paths

$$p(o) = \sum_{\text{reasoning path}} p_f(\text{reasoning path}, o|q)$$

where p_f denotes the probability distribution of Transformer f.

To understand the sample complexity, we focus on the dependence on the following probability gap:

$$\Delta := p(o^*) - \max_{o \in \mathcal{O}, o \neq o^*} p(o)$$

where o^* denotes the correct answer[1]. If $\Delta \leq 0$, then self-consistency fails to find the correct answer with high probability and the separation becomes trivial. Therefore, we focus on the setting where $\Delta > 0$ (i.e., the most likely answer is correct), which is also considered in prior theoretical work (Huang et al., 2024a). Under this setting, we may assume without loss of generality that the reward function r is maximized (only) at the correct answer, because p itself is such a reward function satisfying this condition. Note that since $p(o)$ is marginalized over reasoning paths, $\Delta > 0$ does not imply that the correct answer can be derived easily from greedy decoding.

Theorem 3.1 (Sample Complexity of Self-Consistency)**.** *When $n \geq \frac{2\log(1/\delta)}{\Delta^2}$, self-consistency with n i.i.d. samples is able to produce the correct answer with probability at least $1-\delta$; When $n \leq \frac{1}{\Delta^2}$, there exists a hard instance where self-consistency with n i.i.d. samples fails to produce the correct answer with constant probability.*

Theorem 3.2 (Sample Complexity of Best-of-n)**.** *When $n \geq \frac{2\log(1/\delta)}{\Delta}$, best-of-$n$ with n i.i.d. samples is able to produce the correct answer with probability at least $1-\delta$; When $n \leq \frac{1}{\Delta}$, there exists a hard instance where best-of-n with n i.i.d. samples fails to produce the correct answer with constant probability.*

By providing matching (up to logarithmic factors) upper and lower bounds on the number of samples, the above results establishes the separation between self-consistency and best-of-n. While self-consistency requires $\Theta(1/\Delta^2)$ samples to produce the correct answer, best-of-n shows advantage by only requiring $\Theta(1/\Delta)$ samples. Therefore, this theory corroborates the empirical findings that best-of-n generally leads to better problem solving accuracy on reasoning tasks compared with self-consistency (Sun et al., 2024; Wu et al., 2025a).

4 Expressiveness under Self-Correction

A key distinction between self-correction and the repeated sampling strategies discussed in the previous section lies in the dependence structure of the generated responses: unlike repeated sampling, the outputs produced by self-correction are not i.i.d.. Consequently, to analyze the sample efficiency of self-correction, we must first address a fundamental question: can a large language model (LLM), through self-correction, increase the likelihood of generating the correct answer? At its core, this question is one of expressiveness—whether the Transformer architecture's representation capacity is sufficient to support such improvement.

[1] If there are multiple correct answers, we can let o^* to denote the set, and our results continue to hold in this setting.

In this section, we take a first step toward analyzing the expressiveness of Transformers under the self-correction paradigm. Unlike prior work that focuses on expressiveness in the context of a single task, we study what we call *general-purpose expressiveness*: the ability to solve a broad range of tasks. To this end, we introduce the concept of a General-Purpose Transformer—a construction that maps any collection of task-specific Transformers (experts) into a single unified Transformer.

Definition 4.1 (General-Purpose Transformer)**.** We say that ϕ is a General-Purpose Transformer of type (t_1, t_2) if it maps any set of Transformers with hidden size d and depth L into another 'unified' Transformer with hidden size $t_1 \cdot d + t_2$ and depth $L + O(1)$.

A general-purpose Transformer provides a principled framework for constructing more powerful Transformer architectures by composing simpler, task-specific components. This meta-architecture enables a single model to solve multiple tasks at inference time, representing a significant advancement in our theoretical understanding of the expressive power of modern machine learning systems. Our goal is to investigate the general-purpose expressiveness of self-correction paradigms through the lens of general-purpose Transformers: specifically, how a Transformer can adaptively solve different tasks during inference without prior knowledge of the task identity.

4.1 General-purpose expressiveness

In this section, we present two auxiliary results that serve as building blocks for constructing general-purpose Transformers capable of solving multiple tasks. These results may also be of independent interest beyond expressiveness of self-correction.

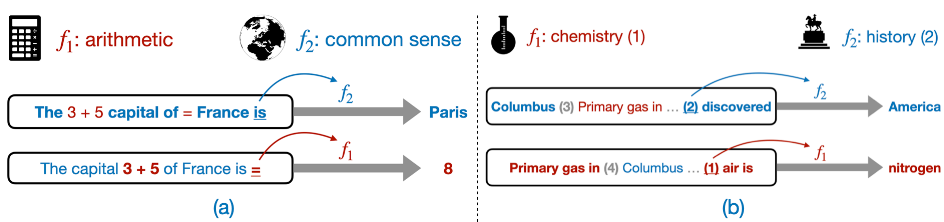

Figure 2: (a): Illustration of Proposition 4.2. In the first query, f_2 is called to solve the common sense problem by attending to only blue tokens. In the second query, f_1 is called to solve the arithmetic problem by attending to only red tokens. (b): Illustration of Proposition 4.4. In the first query, f_2 is called to solve the history problem by attending to only blue tokens. In the second query, f_1 is called to solve the chemistry problem by attending to only red tokens. Importantly, these function calls occur implicitly within the internal computation of the unified Transformer architecture.

The first result addresses the setting in which multiple Transformers operate over distinct vocabularies, with each vocabulary corresponding to a specific task. The objective is to construct a unified Transformer that uses the final token in the input sequence to infer which task to perform, and subsequently solves the task by attending only to the task-relevant tokens.

Proposition 4.2 (General-purpose Expressiveness over Different Token Spaces)**.** *For any $H, L, K, N_{\max} \in \mathbb{Z}_+$, $\mathcal{V}_i \cap \mathcal{V}_j = \emptyset$ $(\forall i \neq j \in \{0\} \cup [K])$, there exists a general-purpose Transformer ϕ of type $(O(K), O(\log N_{\max}))$ such that for any Transformers $f_k = (\theta, \mathrm{pe}, (\mathbf{K}^{(l)}_{k;h}, \mathbf{Q}^{(l)}_{k;h}, \mathbf{V}^{(l)}_{k;h})_{h\in[H],l\in[L]}, \vartheta, \mathcal{V}_k)$ for $k \in [K]$, the Transformer $\widetilde{f} = \phi(f_1, \ldots, f_K)$ satisfies the following property: for any token sequence $v = v_1 \cdots v_n$ such that $n \leq N_{\max}$ and there exists one $v_{i_0} \in \mathcal{V}_0$, we have*

$$p_{\widetilde{f}}(\cdot|v) = p_{f_\kappa}(\cdot|u)$$

where κ is the task indicated by the last token: i.e., $v_n \in \mathcal{V}_\kappa$, and $u = v_{i_1} \cdots v_{i_m}$, where $\{i_1 < \cdots < i_m\} = \{i : v_i \in \mathcal{V}_\kappa\}$, is the sequence of tokens relevant to task κ.

Remark 4.3. *The existence of v_{i_0} which does not belong to any $\{\mathcal{V}_i\}_{i\in[K]}$ serves the technical purpose of inducing attention sink of all irrelevant experts to v_{i_0}. It may be achieved by assuming the user query always ends with the special token* `<eos>`.

The following result considers a more challenging scenario in which multiple Transformers operate across different tasks but share a common vocabulary space. A set of indicator tokens, denoted by

Ω, is used to specify the intended task. The objective is to determine which task to execute based on the most recent indicator token. It then proceeds to solve the task by attending exclusively to the task-relevant tokens appearing before the first indicator token and after the last indicator token in the input sequence.

Proposition 4.4 (Multi-Task Representation over the Same Token Space)**.** *For any $H, L, K, N_{\max} \in \mathbb{Z}_+$, token spaces $\Omega \cap \mathcal{V} = \emptyset$, there exists a general-purpose Transformer ϕ of type $(O(K), O(\log N_{\max}))$ such that for any Transformers $f_k = (\theta, \mathrm{pe}, (\mathbf{K}_{k;h}^{(l)}, \mathbf{Q}_{k;h}^{(l)}, \mathbf{V}_{k;h}^{(l)})_{h\in[H], l\in[L]}, \vartheta, \mathcal{V}), k \in [K]$ over $\mathcal{V}$, the Transformer $\widetilde{f} = \phi(f_1, \ldots, f_K)$ satisfies the following property: for any token sequence $v = v_1 \cdots v_n$ such that*

$$\{\xi_1 < \cdots < \xi_m\} = \{j : v_j \in \Omega\},\ \xi_m < n \leq N_{\max}$$

then we have

$$p_{\widetilde{f}}(\cdot|v) = p_{f_\kappa}(\cdot|u) \tag{2}$$

where $u = v_1 \cdots v_{\xi_1 - 1} v_{\xi_m + 1} \cdots v_n$ is the token sequence obtained by omitting tokens from position ξ_1 to ξ_m, and κ is the task indicated by token v_{ξ_m}.

Remark 4.5. *We observe that in both results above, reducing the type parameters is generally not feasible. The dependence on K arises from the need to compute features for all K experts corresponding to the user query. Since the model lacks prior knowledge of the task, it must encode all task-relevant information to preserve the ability to invoke any expert at inference time. The $\log(N_{\max})$ scaling stems from the positional encoding: in order to construct $N_{\max}$ nearly orthogonal vectors, the positional embedding must have dimension at least $\log(N_{\max})$.*

4.2 General-purpose expressiveness of Transformers with self-correction

In this section we state the main result that establishes general-purpose expressiveness of Transformers with self-correction. We rely on the following notion of regret-minimization Transformer, which expresses the single task of finding the most rewardable action.

Definition 4.6 (Regret-Minimization Transformer)**.** We say that a Transformer f achieves simple regret $\mathrm{reg}(\cdot)$ over reward function r and action space $\mathcal{A}$, if for any $T \in \mathbb{Z}_+$, we have $\max_{a^* \in \mathcal{A}} r(a^*) - \mathbb{E}[r(a_T)] \leq \mathrm{reg}(T)$ where $a_1, \ldots, a_T$ are generated in the following way:

$$\begin{aligned} &a_t \sim p_f(\cdot|a_1, r_1, \ldots, a_{t-1}, r_{t-1}),\ \forall t = 1, \ldots, T, \\ &r_t = r(a_t),\ \forall t = 1, \ldots, T. \end{aligned}$$

Essentially, the goal of a regret-minimization Transformer is to learn from a reward oracle and ultimately recommend an action that is near-optimal, which is related to a concept commonly referred to as simple regret in the bandit literature (Even-Dar et al., 2006; Carpentier & Valko, 2015; Jamieson et al., 2014). To achieve this, the Transformer may implement strategies such as mirror descent, upper confidence bounds, or search-based algorithms, depending on the problem structure. As these procedures rely only on basic arithmetic operations, such Transformers can be constructed by applying the universal approximation capabilities of Transformers (Yun et al., 2020; Luo et al., 2022; Feng et al., 2023; Li et al., 2024b): For example, Lin et al. (2023) provide constructions to approximate upper confidence bounds and Thompson sampling algorithms with regret $O(\sqrt{T})$. Consequently, their construction is not the primary focus of this work.

The following theorem establishes the existence of a general-purpose Transformer that can simulate the behavior of a set of expert Transformers (not necessarily over the same token space) through self-correction. Specifically, it shows that such a unified Transformer can, at inference time, identify and invoke the appropriate expert to solve any task that the original experts can solve. The self-correction protocol is described in Algorithm 1, wherein the unified Transformer autoregressively generates actions and responses, after which the verifier is queried to obtain reward signals. Through this process of trial and error, the model effectively "learns" at inference time, using the verifier to minimize regret and adaptively select the correct expert.

Theorem 4.7 (Regret Minimization via Self-Correction)**.** *For any $H, L, K, N_{\max} \in \mathbb{Z}_+$, token spaces $\mathcal{V}_0, \mathcal{V}_1, \ldots, \mathcal{V}_K, \mathcal{A}$ ($|\mathcal{A}| = K$) such that $\mathcal{V}_0, \mathcal{V} = (\cup_{k=1}^K \mathcal{V}_k)$, and $\mathcal{A}$ are disjoint, and reward function r, there exists a general-purpose Transformer ϕ of type $(O(K), O(\log N_{\max}))$ such that given any set of Transformers denoted as follows,*

Algorithm 1 Self-correction with verifier

```
1: procedure GENERATION(q)                                  ▷ q = q_1 ... q_{n_0} denotes the user query.
2:     prompt ← q
3:     for t = 1, ..., T do
4:         a^(t) ~ p_f̃(· | prompt)                          ▷ a^(t) designates which expert to use in t-th iteration
5:         prompt ← prompt|a^(t)    ▷ Update the prompt autoregressively, | represents token concatenation.
6:         for i = 1, ... do
7:             u_i^(t) ~ p_f̃(· | prompt)                    ▷ Generate t-th response autoregressively
8:             prompt ← prompt|u_i^(t)                       ▷ Update the prompt autoregressively
9:             if u_i^(t) = EOS then
10:                Break
11:        r^(t) ← r(q, u^(t)), prompt ← prompt|r^(t)       ▷ Query verifier to obtain reward of t-th response
12:    Return
```

- *K **expert Transformers**: $f_k = (\theta, \mathrm{pe}, (\mathbf{K}^{(l)}_{k;h}, \mathbf{Q}^{(l)}_{k;h}, \mathbf{V}^{(l)}_{k;h})_{h\in[H], l\in[L]}, \vartheta, \mathcal{V}_k)$ for $k \in [K]$, such that one of the expert f_{k^*} achieves λ-suboptimal reward:*
$$\mathbb{E}_{u\sim f_{k^*}(\cdot|q)}[r(q,u)] \geq \max_{u^*\in\mathcal{V}^\omega} r(q,u^*) - \lambda$$
- ***Regret-Minimization Transformer**: $f_0 = (\theta, \mathrm{pe}, \mathbf{K}^{(l)}_{0;h}, \mathbf{Q}^{(l)}_{0;h}, \mathbf{V}^{(l)}_{0;h})_{h\in[H], l\in[L]}, \vartheta, \mathcal{V}_0 \cup \mathcal{A})$ that implements a bandit algorithm over the reward function r_0 and action space $\mathcal{A}$ with simple regret $\mathrm{reg}(t)$, where $r_0(a) = \mathbb{E}_{u\sim f_a(\cdot|q)}[r(q,u)]$ denotes the average reward of responses generated by the a-th expert,*

then the Transformer $\widetilde{f} = \phi(f_0, f_1, \ldots, f_K)$ satisfies the following property: for any prompt $v = v_1 \cdots v_n$, if the response sequence $u^{(1)}, \ldots, u^{(T)}$ generated by the protocol in Algorithm 1 has total length $\leq N_{\max}$, then we have
$$\max_{u^*\in\mathcal{V}^\omega} r(q,u^*) - \mathbb{E}[r(q,u^{(T)})] \leq \lambda + \mathrm{reg}(T)$$

Remark 4.8. *While the general-purpose Transformer ϕ can be applied to construct the brutal-force Transformer $\widetilde{f}$ that simply tries every expert, we note that the generality of Definition 4.6 allows us to construct more powerful Transformers beyond brutal search. Leveraging the structures in the problem and the expert pool, it is entirely possible to identify the correct expert using $\ll K$ trials (Russo & Van Roy, 2018; Foster et al., 2021).*

As a consequence of Theorem 4.7, we obtain a Transformer architecture that can provably produce a final answer that nearly maximizes the reward. This means that the unified transformer can solve K distinct tasks at inference time, without requiring prior knowledge of which task the user query pertains to. Notably, the construction of such an architecture is *general-purpose*, in that it is independent of the specific tasks, reward functions, or expert policies. To the best of our knowledge, this constitutes the first theoretical result of its kind in the study of Transformer architectures. Furthermore, our theory aligns with the empirical finding that LLMs are able to progressively optimize outcome rewards during test-time (Qu et al., 2025; Song et al., 2025; Team, 2025; Monea et al., 2024).

5 Experiments

In this section, we conduct synthetic experiments to show that Transformers can self-correct with verifier feedback.

5.1 Expressiveness of Self-Correction

Data generation. We aim to construct a test problem with complex prompts such that correctly solving the problem in the single-term generation is challenging. In this case, self-correction can play a critical role if Transformers have such capacities. Specifically, in our synthetic problem, the prompt is the concatenation of the following two components:

- **Instruction**: A 3-SAT problem, e.g.,

$$(\sim x_3 \vee \sim x_1 \vee \sim x_2) \wedge (\sim x_1 \vee \sim x_3 \vee x_2) \wedge (\sim x_4 \vee x_2 \vee \sim x_3) \wedge \cdots$$

- **Data**: A string composed of characters from the set $\{\texttt{a}, \texttt{b}\}$.

The ground truth target is defined as follows: If the 3-SAT problem in the *instruction* is satisfiable, the model should *copy* the string in the *data* part in the output; otherwise, the model should *reverse* the string in the output. In our experiment, we construct datasets using 3-SAT problems with 4 variables and 20 clauses. The lengths of the data strings are set to 5. We generate 10000 instances for training and 512 instances for evaluation. In the training set, we control the ratio of satisfiable and unsatisfiable 3-SAT instructions to 9:1, while in the test set, the ratio is set to 1:1. This label imbalance ensures that models fail to answer the question correctly in the first attempt and thus elicit the self-correction behavior.

Models and training configuration. We train a class of Transformer models of various sizes: {GPT-nano, GPT-micro, GPT-mini, Gopher-44M} with the Adam optimizer Kingma & Ba (2015) for 5 epochs. More implementation details can be found in Appendix B.

Results. Test set accuracy across different inference settings is shown in Figure 3. We note that model performance plateaus at 63.19% when there is no self-correction at test time, with no improvement from increased model size. By contrast, when models are equipped with verifier signals to enable self-correction, test accuracy improves substantially, demonstrating the efficacy of this mechanism. Crucially, larger models – such as GPT-mini and Gopher-44M – achieve near-perfect accuracy under self-correction, suggesting that sufficiently expressive Transformers are capable of implementing effective self-correction strategies. This empirical result supports our theoretical findings.

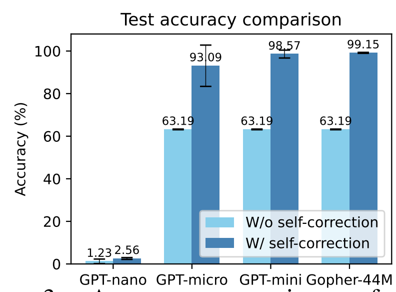

Figure 3: Accuracy comparisons of different models with/without self-correction at test time.

5.2 Evaluation of Sample Complexity

Dataset. We conduct experiments on the AIME 2024 & 2025 datasets (Mathematical Association of America, 2025), which serve as a real-world benchmark for evaluating mathematical reasoning tasks. These datasets allow us to measure not only the raw accuracy of different large language models (LLMs), but also the impact of verification-based strategies on sample efficiency.

Model configuration. We consider recent LLMs, including `Qwen3-1.7B`, `Qwen3-4B` (Yang et al., 2025), and `Llama-3.2-3B-Instruct` (Dubey et al., 2024), as candidate models. In addition, `Qwen3-32B` is employed as an LLM verifier. This setup enables us to compare standard decoding strategies (self-consistency) with verification-based methods (best-of and self-correction).

Model \ Method	**Self-consistency (64 samples)**	**Best-of-n (4 samples)**	**Self-correction (4 samples)**
Qwen3-1.7B	58.33%	59.68%	79.29%
Qwen3-4B	78.33%	80.58%	81.19%
Llama-3.2-3B-Instruct	1.67%	4.84%	24.52%

Table 1: Accuracy comparison of self-consistency, best-of-n, and self-correction methods on AIME 24 & 25 datasets.

Results. We compare the accuracy of self-consistency, best-of-n, and self-correction under different sample sizes. Notably, as summarized in Table 1, best-of with only 4 samples consistently outperforms self-consistency with 64 samples, confirming the predicted gap in sample complexity. Furthermore, self-correction with verifiers achieves strong performance, highlighting the ability of LLMs to leverage verifier feedback effectively. These results show a notable sample complexity gap between Self-consistency and Best-of-n and confirm that modern Transformer models are sufficiently expressive to implement self-correction mechanisms when combined with verifiers, thus validating our theoretical results in Section 3 and 4.

6 Related Works

Theories of Transformers and Large Language Models. The success of Transformers and LLMs has motivated the study on their expressiveness. Existing research has shown that Transformers can implement simple functions such as sparse linear functions, two-layer neural networks, and decision trees (Garg et al., 2022), gradient descent (Akyürek et al., 2022; Bai et al., 2023; Von Oswald et al., 2023), automata (Liu et al., 2022; Zhao et al., 2023), Dyck languages (Bhattamishra et al., 2020a; Yao et al., 2021), Turing machines (Dehghani et al., 2018; Bhattamishra et al., 2020b; Zaheer et al., 2020; Pérez et al., 2021; Wei et al., 2022a), variational inference (Mei & Wu, 2023), and bandit algorithms (Lin et al., 2023). Yun et al. (2020); Luo et al. (2022); Alberti et al. (2023); Petrov et al. (2024) establish universal approximation results under various settings. Edelman et al. (2022); Elhage et al. (2021); Li et al. (2021); Likhosherstov et al. (2021) study representational capabilities and properties of self-attention, the core component in Transformers. Feng et al. (2023); Li et al. (2024b) study the expressiveness of auto-regressive Transformers with chain-of-thought. Edelman et al. (2022); Li et al. (2024a); Botta et al. (2025) studies the sample complexity of Transformers. Recently, a growing body of work has begun to explore the theoretical foundations of self-improvement in large language models (LLMs). Song et al. (2024b) introduces the generation-verification gap as a key quantity governing scaling behavior. Huang et al. (2024a) proposes a progressive sharpening framework in which the policy gradually shifts toward more confident responses. Setlur et al. (2025) draws on reinforcement learning theory to formally establish the advantages of verifier-based methods. In contrast to these works, our results provide explicit sample complexity rates and tangible representation architectures, enabling a more concrete understanding of the fundamental capabilities and limitations of test-time scaling paradigms.

Test-time scaling. Recent research has established the test-time scaling law of LLMs, illuminating a new scaling axis beyond training-time scaling laws (Kaplan et al., 2020; Hoffmann et al., 2022). Existing approaches of scaling up test-time compute of LLMs can be broadly classified into two categories: (1) applying test-time algorithms (aka inference-time algorithms) during LLM decoding (Brown et al., 2024; Wu et al., 2025a; Snell et al., 2025); and (2) explicitly training LLMs to output long chain-of-thought traces (Guo et al., 2025; Kimi, 2025; OpenAI, 2024b; Yang et al., 2025). Many recent works focus on understanding and improving the effectiveness of test-time scaling empirically: Chen et al. (2024c); Aggarwal & Welleck (2025); Cuadron et al. (2025); Wang et al. (2025) study under-thinking, over-thinking, and length control in LLM reasoning. Chen et al. (2025) proposes to integrate self-verification and self-correction into sampling. Qu et al. (2025) analyze optimizing test-time compute by introducing a meta reinforcement learning formulation. Setlur et al. (2025) demonstrate that verification/RL is important for optimal test-time scaling. Zhang et al. (2025) provide an extensive review of the test-time scaling landscape. In contrast, our work focuses on theoretical analyses of test-time scaling. In addition, our work provides theoretical explanation of In-Context Reinforcement Learning (Song et al., 2025; Team, 2025; Monea et al., 2024).

7 Discussions

Our investigation reveals a fundamental separation in sample complexity between self-consistency and best-of-n, providing theoretical support for the empirically observed superiority of the latter method. Furthermore, by introducing the framework of *general-purpose expressiveness*, we construct generic Transformer architectures capable of emulating online learning algorithms at test time. This capability enables a single model to provably solve multiple tasks without task-specific adaptation, thus extending our understanding of expressiveness to multi-task settings. Our experiments validate the theoretical separation and confirms that it requires additional model capacities for Transformer to implement self-correction.

References

Pranjal Aggarwal and Sean Welleck. L1: Controlling how long a reasoning model thinks with reinforcement learning. In *arXiv*, 2025. URL https://arxiv.org/abs/2503.04697.

Shipra Agrawal and Randy Jia. Optimistic posterior sampling for reinforcement learning: worst-case regret bounds. *Advances in neural information processing systems*, 30, 2017.

Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. What learning algorithm is in-context learning? investigations with linear models. *arXiv preprint arXiv:2211.15661*, 2022.

Silas Alberti, Niclas Dern, Laura Thesing, and Gitta Kutyniok. Sumformer: Universal approximation for efficient transformers. In Timothy Doster, Tegan Emerson, Henry Kvinge, Nina Miolane, Mathilde Papillon, Bastian Rieck, and Sophia Sanborn (eds.), *Proceedings of 2nd Annual Workshop on Topology, Algebra, and Geometry in Machine Learning (TAG-ML)*, volume 221 of *Proceedings of Machine Learning Research*, pp. 72–86. PMLR, 28 Jul 2023. URL https://proceedings.mlr.press/v221/alberti23a.html.

Cem Anil, Yuhuai Wu, Anders Andreassen, Aitor Lewkowycz, Vedant Misra, Vinay Ramasesh, Ambrose Slone, Guy Gur-Ari, Ethan Dyer, and Behnam Neyshabur. Exploring length generalization in large language models. *arXiv preprint arXiv:2207.04901*, 2022.

Yu Bai, Fan Chen, Huan Wang, Caiming Xiong, and Song Mei. Transformers as statisticians: Provable in-context learning with in-context algorithm selection. *arXiv preprint arXiv:2306.04637*, 2023.

Boaz Barak, Benjamin Edelman, Surbhi Goel, Sham Kakade, Eran Malach, and Cyril Zhang. Hidden progress in deep learning: Sgd learns parities near the computational limit. *Advances in Neural Information Processing Systems*, 35:21750–21764, 2022.

Satwik Bhattamishra, Kabir Ahuja, and Navin Goyal. On the ability and limitations of transformers to recognize formal languages. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 7096–7116, 2020a.

Satwik Bhattamishra, Arkil Patel, and Navin Goyal. On the computational power of transformers and its implications in sequence modeling. In *Proceedings of the 24th Conference on Computational Natural Language Learning*, pp. 455–475, 2020b.

Edoardo Botta, Yuchen Li, Aashay Mehta, Jordan T Ash, Cyril Zhang, and Andrej Risteski. On the query complexity of verifier-assisted language generation. *arXiv preprint arXiv:2502.12123*, 2025.

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V Le, Christopher Ré, and Azalia Mirhoseini. Large language monkeys: Scaling inference compute with repeated sampling. *arXiv preprint arXiv:2407.21787*, 2024.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*, 2023.

Alexandra Carpentier and Michal Valko. Simple regret for infinitely many armed bandits. In *International Conference on Machine Learning*, pp. 1133–1141. PMLR, 2015.

Guoxin Chen, Minpeng Liao, Chengxi Li, and Kai Fan. Alphamath almost zero: Process supervision without process. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024a. URL https://openreview.net/forum?id=VaXnxQ3UKo.

Jiefeng Chen, Jie Ren, Xinyun Chen, Chengrun Yang, Ruoxi Sun, and Sercan Ö Arık. Sets: Leveraging self-verification and self-correction for improved test-time scaling. *arXiv preprint arXiv:2501.19306*, 2025.

Lingjiao Chen, Jared Quincy Davis, Boris Hanin, Peter Bailis, Ion Stoica, Matei Zaharia, and James Zou. Are more LLM calls all you need? towards the scaling properties of compound AI systems. In *Conference on Neural Information Processing Systems*, 2024b. URL https://openreview.net/forum?id=m5106RRLgx.

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, et al. Do not think that much for 2+ 3=? on the overthinking of o1-like llms. *arXiv preprint arXiv:2412.21187*, 2024c.

Xinyun Chen, Maxwell Lin, Nathanael Schärli, and Denny Zhou. Teaching large language models to self-debug. In *International Conference on Learning Representations*, 2024d. URL https://openreview.net/forum?id=KuPixIqPiq.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.

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

codeforce. Codeforces, 2025. URL https://codeforces.com/.

Alejandro Cuadron, Dacheng Li, Wenjie Ma, Xingyao Wang, Yichuan Wang, Siyuan Zhuang, Shu Liu, Luis Gaspar Schroeder, Tian Xia, Huanzhi Mao, et al. The danger of overthinking: Examining the reasoning-action dilemma in agentic tasks. *arXiv preprint arXiv:2502.08235*, 2025.

DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. In *arXiv*, 2025. URL https://arxiv.org/abs/2501.12948.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Łukasz Kaiser. Universal transformers. *arXiv preprint arXiv:1807.03819*, 2018.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv e-prints*, pp. arXiv–2407, 2024.

Benjamin L Edelman, Surbhi Goel, Sham Kakade, and Cyril Zhang. Inductive biases and variable creation in self-attention mechanisms. In *International Conference on Machine Learning*, pp. 5793–5831. PMLR, 2022.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, et al. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 1:1, 2021.

Eyal Even-Dar, Shie Mannor, Yishay Mansour, and Sridhar Mahadevan. Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. *Journal of machine learning research*, 7(6), 2006.

Guhao Feng, Bohang Zhang, Yuntian Gu, Haotian Ye, Di He, and Liwei Wang. Towards revealing the mystery behind chain of thought: a theoretical perspective. *Advances in Neural Information Processing Systems*, 36:70757–70798, 2023.

Dylan J Foster, Sham M Kakade, Jian Qian, and Alexander Rakhlin. The statistical complexity of interactive decision making. *arXiv preprint arXiv:2112.13487*, 2021.

Zitian Gao, Boye Niu, Xuzheng He, Haotian Xu, Hongzhang Liu, Aiwei Liu, Xuming Hu, and Lijie Wen. Interpretable contrastive monte carlo tree search reasoning. In *arXiv*, 2024. URL https://arxiv.org/abs/2410.01707.

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

Olga Golovneva, Tianlu Wang, Jason Weston, and Sainbayar Sukhbaatar. Contextual position encoding: Learning to count what's important. *arXiv preprint arXiv:2405.18719*, 2024.

Zhibin Gou, Zhihong Shao, Yeyun Gong, yelong shen, Yujiu Yang, Nan Duan, and Weizhu Chen. CRITIC: Large language models can self-correct with tool-interactive critiquing. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=Sx038qxjek.

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

Zhenyu He, Guhao Feng, Shengjie Luo, Kai Yang, Liwei Wang, Jingjing Xu, Zhi Zhang, Hongxia Yang, and Di He. Two stones hit one bird: Bilevel positional encoding for better length extrapolation. *arXiv preprint arXiv:2401.16421*, 2024.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2021. URL https://openreview.net/forum?id=7Bywt2mQsCe.

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, Katherine Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Oriol Vinyals, Jack William Rae, and Laurent Sifre. An empirical analysis of compute-optimal large language model training. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=iBBcRUlOAPR.

Audrey Huang, Adam Block, Dylan J Foster, Dhruv Rohatgi, Cyril Zhang, Max Simchowitz, Jordan T Ash, and Akshay Krishnamurthy. Self-improvement in language models: The sharpening mechanism. *arXiv preprint arXiv:2412.01951*, 2024a.

Zhen Huang, Zengzhi Wang, Shijie Xia, Xuefeng Li, Haoyang Zou, Ruijie Xu, Run-Ze Fan, Lyumanshan Ye, Ethan Chern, Yixin Ye, Yikai Zhang, Yuqing Yang, Ting Wu, Binjie Wang, Shichao Sun, Yang Xiao, Yiyuan Li, Fan Zhou, Steffi Chern, Yiwei Qin, Yan Ma, Jiadi Su, Yixiu Liu, Yuxiang Zheng, Shaoting Zhang, Dahua Lin, Yu Qiao, and Pengfei Liu. Olympicarena: Benchmarking multi-discipline cognitive reasoning for superintelligent AI. In *Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2024b. URL https://openreview.net/forum?id=ayF8bEKYQy.

Robert Irvine, Douglas Boubert, Vyas Raina, Adian Liusie, Ziyi Zhu, Vineet Mudupalli, Aliaksei Korshuk, Zongyi Liu, Fritz Cremer, Valentin Assassi, Christie-Carol Beauchamp, Xiaoding Lu, Thomas Rialan, and William Beauchamp. Rewarding chatbots for real-world engagement with millions of users. In *arXiv*, 2023. URL https://arxiv.org/abs/2303.06135.

Kevin Jamieson, Matthew Malloy, Robert Nowak, and Sébastien Bubeck. lilucb: An optimal exploration algorithm for multi-armed bandits. In *Conference on Learning Theory*, pp. 423–439. PMLR, 2014.

Nirmit Joshi, Gal Vardi, Adam Block, Surbhi Goel, Zhiyuan Li, Theodor Misiakiewicz, and Nathan Srebro. A theory of learning with autoregressive chain of thought. *arXiv preprint arXiv:2503.07932*, 2025.

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

Kimi. Kimi k1.5: Scaling reinforcement learning with llms. In *arXiv*, 2025. URL https://arxiv.org/abs/2501.12599.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR (Poster)*, 2015.

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

Shanda Li, Xiangning Chen, Di He, and Cho-Jui Hsieh. Can vision transformers perform convolution? *arXiv preprint arXiv:2111.01353*, 2021.

Shanda Li, Tanya Marwah, Junhong Shen, Weiwei Sun, Andrej Risteski, Yiming Yang, and Ameet Talwalkar. Codepde: An inference framework for llm-driven pde solver generation. *arXiv preprint arXiv:2505.08783*, 2025.

Shuai Li, Zhao Song, Yu Xia, Tong Yu, and Tianyi Zhou. The closeness of in-context learning and weight shifting for softmax regression. *arXiv preprint arXiv:2304.13276*, 2023.

Yuchen Li, Alexandre Kirchmeyer, Aashay Mehta, Yilong Qin, Boris Dadachev, Kishore Papineni, Sanjiv Kumar, and Andrej Risteski. Promises and pitfalls of generative masked language modeling: theoretical framework and practical guidelines. *arXiv preprint arXiv:2407.21046*, 2024a.

Zhiyuan Li, Hong Liu, Denny Zhou, and Tengyu Ma. Chain of thought empowers transformers to solve inherently serial problems. In *The Twelfth International Conference on Learning Representations*, 2024b. URL https://openreview.net/forum?id=3EWTEy9MTM.

Valerii Likhosherstov, Krzysztof Choromanski, and Adrian Weller. On the expressive power of self-attention matrices. *arXiv preprint arXiv:2106.03764*, 2021.

Licong Lin, Yu Bai, and Song Mei. Transformers as decision makers: Provable in-context reinforcement learning via supervised pretraining. *arXiv preprint arXiv:2310.08566*, 2023.

Qingwen Lin, Boyan Xu, Zijian Li, Zhifeng Hao, Keli Zhang, and Ruichu Cai. Leveraging constrained monte carlo tree search to generate reliable long chain-of-thought for mathematical reasoning. In *arXiv*, 2025. URL https://arxiv.org/abs/2502.11169.

Bingbin Liu, Jordan T Ash, Surbhi Goel, Akshay Krishnamurthy, and Cyril Zhang. Transformers learn shortcuts to automata. *arXiv preprint arXiv:2210.10749*, 2022.

Shengjie Luo, Shanda Li, Shuxin Zheng, Tie-Yan Liu, Liwei Wang, and Di He. Your transformer may not be as powerful as you expect. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=NQFFNdsOGD.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=S37hOerQLB.

Mathematical Association of America. American invitational mathematics examination. https://maa.org/maa-invitational-competitions, 2025.

Song Mei and Yuchen Wu. Deep networks as denoising algorithms: Sample-efficient learning of diffusion models in high-dimensional graphical models. *arXiv preprint arXiv:2309.11420*, 2023.

William Merrill and Ashish Sabharwal. The expressive power of transformers with chain of thought. *arXiv preprint arXiv:2310.07923*, 2023.

Giovanni Monea, Antoine Bosselut, Kianté Brantley, and Yoav Artzi. Llms are in-context bandit reinforcement learners. *arXiv preprint arXiv:2410.05362*, 2024.

Tergel Munkhbat, Namgyu Ho, Seo Hyun Kim, Yongjin Yang, Yujin Kim, and Se-Young Yun. Self-training elicits concise reasoning in large language models. In *arXiv*, 2025. URL https://arxiv.org/abs/2502.20122.

Alex Nguyen, Dheeraj Mekala, Chengyu Dong, and Jingbo Shang. When is the consistent prediction likely to be a correct prediction? In *arXiv*, 2024. URL https://arxiv.org/abs/2407.05778.

Catherine Olsson, Nelson Elhage, Neel Nanda, Nicholas Joseph, Nova DasSarma, Tom Henighan, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, et al. In-context learning and induction heads. *arXiv preprint arXiv:2209.11895*, 2022.

OpenAI. Openai o1 system card. In *arXiv*, 2024a. URL https://arxiv.org/abs/2412.16720.

OpenAI. Openai o3-mini, 2024b. URL https://openai.com/index/openai-o3-mini/.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32: 8026–8037, 2019.

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. In *Proceedings of the 41st International Conference on Machine Learning*, pp. 40523–40550, 2024.

Jiahao Qiu, Yifu Lu, Yifan Zeng, Jiacheng Guo, Jiayi Geng, Huazheng Wang, Kaixuan Huang, Yue Wu, and Mengdi Wang. Treebon: Enhancing inference-time alignment with speculative tree-search and best-of-n sampling. *arXiv preprint arXiv:2410.16033*, 2024.

Yuxiao Qu, Matthew YR Yang, Amrith Setlur, Lewis Tunstall, Edward Emanuel Beeching, Ruslan Salakhutdinov, and Aviral Kumar. Optimizing test-time compute via meta reinforcement fine-tuning. *arXiv preprint arXiv:2503.07572*, 2025.

Daniel Russo and Benjamin Van Roy. Learning to optimize via information-directed sampling. *Operations Research*, 66(1):230–252, 2018.

Pier Giuseppe Sessa, Robert Dadashi, Léonard Hussenot, Johan Ferret, Nino Vieillard, Alexandre Ramé, Bobak Shariari, Sarah Perrin, Abe Friesen, Geoffrey Cideron, Sertan Girgin, Piotr Stanczyk, Andrea Michi, Danila Sinopalnikov, Sabela Ramos, Amélie Héliou, Aliaksei Severyn, Matt Hoffman, Nikola Momchev, and Olivier Bachem. Bond: Aligning llms with best-of-n distillation. In *arXiv*, 2024. URL https://arxiv.org/abs/2407.14622.

Amrith Setlur, Nived Rajaraman, Sergey Levine, and Aviral Kumar. Scaling test-time compute without verification or rl is suboptimal. *arXiv preprint arXiv:2502.12118*, 2025.

Ben Shi, Michael Tang, Karthik R Narasimhan, and Shunyu Yao. Can language models solve olympiad programming? In *Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=kGa4fMtP9l.

Charlie Victor Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=4FWAwZtd2n.

Kefan Song, Amir Moeini, Peng Wang, Lei Gong, Rohan Chandra, Yanjun Qi, and Shangtong Zhang. Reward is enough: Llms are in-context reinforcement learners. *arXiv preprint arXiv:2506.06303*, 2025.

Yifan Song, Guoyin Wang, Sujian Li, and Bill Yuchen Lin. The good, the bad, and the greedy: Evaluation of llms should not ignore non-determinism. In *arXiv*, 2024a. URL https://arxiv.org/abs/2407.10457.

Yuda Song, Hanlin Zhang, Carson Eisenach, Sham Kakade, Dean Foster, and Udaya Ghai. Mind the gap: Examining the self-improvement capabilities of large language models. *arXiv preprint arXiv:2412.02674*, 2024b.

Zhiqing Sun, Longhui Yu, Yikang Shen, Weiyang Liu, Yiming Yang, Sean Welleck, and Chuang Gan. Easy-to-hard generalization: Scalable alignment beyond human supervision. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=qwgfh2fTtN.

Tooliense Team. Crux: Autonomous mathematical research through hierarchical multi-agent orchestration, 2025. URL https://github.com/tooliense/crux. IC-RL Implementation with Self-Evolve Mechanism.

Ye Tian, Baolin Peng, Linfeng Song, Lifeng Jin, Dian Yu, Lei Han, Haitao Mi, and Dong Yu. Toward self-improvement of LLMs via imagination, searching, and criticizing. In *Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=tPdJ2qHkOB.

Johannes Von Oswald, Eyvind Niklasson, Ettore Randazzo, João Sacramento, Alexander Mordvintsev, Andrey Zhmoginov, and Max Vladymyrov. Transformers learn in-context by gradient descent. *arXiv preprint arXiv:2212.07677*, 2022.

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

Ziyu Wan, Xidong Feng, Muning Wen, Stephen Marcus McAleer, Ying Wen, Weinan Zhang, and Jun Wang. Alphazero-like tree-search can guide large language model decoding and training. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=C4OpREezgj.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=1PL1NIMMrw.

Yue Wang, Qiuzhi Liu, Jiahao Xu, Tian Liang, Xingyu Chen, Zhiwei He, Linfeng Song, Dian Yu, Juntao Li, Zhuosheng Zhang, et al. Thoughts are all over the place: On the underthinking of o1-like llms. *arXiv preprint arXiv:2501.18585*, 2025.

Colin Wei, Yining Chen, and Tengyu Ma. Statistically meaningful approximation: a case study on approximating turing machines with transformers. *Advances in Neural Information Processing Systems*, 35:12071–12083, 2022a.

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, 2022b. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html.

Sean Welleck, Ximing Lu, Peter West, Faeze Brahman, Tianxiao Shen, Daniel Khashabi, and Yejin Choi. Generating sequences by learning to self-correct. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=hH36JeQZDaO.

Yangzhen Wu, Zhiqing Sun, Shanda Li, Sean Welleck, and Yiming Yang. Scaling inference computation: Compute-optimal inference for problem-solving with language models. In *Workshop on Mathematical Reasoning and AI at NeurIPS'24*, 2024. URL https://openreview.net/forum?id=j7DZWSc8qu.

Yangzhen Wu, Zhiqing Sun, Shanda Li, Sean Welleck, and Yiming Yang. Inference scaling laws: An empirical analysis of compute-optimal inference for LLM problem-solving. In *The Thirteenth International Conference on Learning Representations*, 2025a. URL https://openreview.net/forum?id=VNckp7JEHn.

Yuyang Wu, Yifei Wang, Tianqi Du, Stefanie Jegelka, and Yisen Wang. When more is less: Understanding chain-of-thought length in llms. In *arXiv*, 2025b. URL https://arxiv.org/abs/2502.07266.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

Shunyu Yao, Binghui Peng, Christos Papadimitriou, and Karthik Narasimhan. Self-attention networks can process bounded hierarchical languages. *arXiv preprint arXiv:2105.11115*, 2021.

Chulhee Yun, Srinadh Bhojanapalli, Ankit Singh Rawat, Sashank Reddi, and Sanjiv Kumar. Are transformers universal approximators of sequence-to-sequence functions? In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=ByxRM0Ntvr.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. Big bird: Transformers for longer sequences. *Advances in neural information processing systems*, 33:17283–17297, 2020.

Dan Zhang, Sining Zhoubian, Ziniu Hu, Yisong Yue, Yuxiao Dong, and Jie Tang. ReST-MCTS*: LLM self-training via process reward guided tree search. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024a. URL https://openreview.net/forum?id=8rcFOqEud5.

Kechi Zhang, Ge Li, Huangzhao Zhang, and Zhi Jin. Hirope: Length extrapolation for code models using hierarchical position. *arXiv preprint arXiv:2403.19115*, 2024b.

Qiyuan Zhang, Fuyuan Lyu, Zexu Sun, Lei Wang, Weixu Zhang, Zhihan Guo, Yufei Wang, Irwin King, Xue Liu, and Chen Ma. What, how, where, and how well? a survey on test-time scaling in large language models. *arXiv preprint arXiv:2503.24235*, 2025.

Yunxiang Zhang, Muhammad Khalifa, Lajanugen Logeswaran, Jaekyeom Kim, Moontae Lee, Honglak Lee, and Lu Wang. Small language models need strong verifiers to self-correct reasoning. In *ACL (Findings)*, 2024c. URL https://aclanthology.org/2024.findings-acl.924/.

Yuxiang Zhang, Shangxi Wu, Yuqi Yang, Jiangming Shu, Jinlin Xiao, Chao Kong, and Jitao Sang. o1-coder: an o1 replication for coding. In *arXiv*, 2024d. URL https://arxiv.org/abs/2412.00154.

Haoyu Zhao, Abhishek Panigrahi, Rong Ge, and Sanjeev Arora. Do transformers parse while predicting the masked word? In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 16513–16542, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.1029. URL https://aclanthology.org/2023.emnlp-main.1029/.

A PROOFS

A.1 PROOF OF THEOREM 3.1

Proof. Write $\mathcal{O} = \{1, \ldots, O\}$ ($O \in \mathbb{Z}_+$) where i is the i-th most likely answer and let n_i denote the number of occurrences of i. Then we have

$$\hat{p} = \frac{1}{n}(n_1, \ldots, n_O) \sim \frac{1}{n}\text{Multinomial}(n, p),$$

where $p = (p(1), \ldots, p(O))$.

Upper bound. When $n \geq \frac{2\log(1/\delta)}{\Delta^2}$ we apply Claim A.5 to obtain that with probability at least $1 - \delta$,

$$\|\hat{p} - p\|_1 \leq \sqrt{\frac{2\ln(1/\delta)}{n}} \leq \Delta.$$

Under this event, we have that for any $i > 1$

$$\begin{aligned} n_1 - n_i &= n \cdot (\hat{p}_1 - \hat{p}_i) \\ &\geq n \cdot (p_1 - p_i - \|\hat{p} - p\|_1) \\ &\geq 0 \end{aligned}$$

and hence the correct answer 1 is the most consistent answer. It follows that self-consistency can produce the correct answer with probability at least $1 - \delta$.

Lower bound. When $n \leq \frac{1}{\Delta^2}$, we construct the hard instance where $p_1 = (1 + \Delta)/2, p_2 = (1 - \Delta)/2$ and $\Delta < 0.00001$. If $n \leq \frac{1}{\Delta}$ then by the proof of Theorem 3.2, with constant probability the correct answer is not generated at all and hence self-consistency fails to produce the correct answer. Otherwise $n \geq \frac{1}{\Delta} \geq 10000$. We may write $X := \frac{n_1 - n_2 - n\Delta}{\sqrt{n}}$ as a sum of i.i.d. random variables divided by $\sqrt{n}$:

$$X = \frac{\sum_{i=1}^{n} Y_i}{\sqrt{n}},$$

where $\mathbb{E}(Y_i) = 0, \sigma^2 = \mathbb{E}(Y_i^2) \geq 1/2, \rho = \mathbb{E}(|Y_i|^3) \leq 1$. By Claim A.6, we have that

$$\begin{aligned} \mathbb{P}(n_1 < n_2) &= \mathbb{P}(X < -1) \\ &\geq \Phi(-1) - \frac{8\rho}{\sigma^3\sqrt{n}} \\ &\geq 0.01. \end{aligned}$$

Thus in both cases, self-consistency fails to produce the correct answer with constant probability. □

A.2 PROOF OF THEOREM 3.2

Proof. Write $\mathcal{O} = \{1, \ldots, O\}$ where i is the i-th most likely answer and let n_i denote the number of occurrences of i. Then we have

$$p(1) \geq p(2) + \Delta \geq \Delta.$$

Note that for best-of-n, correctness is achieved if the correct answer appears at least once among n independent samples.

Upper bound. When $n \geq \frac{2\log(1/\delta)}{\Delta}$, we have

$$\begin{aligned} \mathbb{P}(\text{Best-of-}n \text{ outputs correct answer}) &= 1 - (1 - p(1))^n \\ &\geq 1 - (1 - \Delta)^{\frac{2\log(1/\delta)}{\Delta}} \\ &\geq 1 - \delta. \end{aligned}$$

This confirms that best-of-n achieves the correct answer with $1 - \delta$ probability.

Lower bound. When $n \le \frac{1}{\Delta}$, we construct the hard instance where $p(1) = \Delta + (1-\Delta)/O, p(2) = \cdots = p(O) = (1-\Delta)/O$ and $\Delta < 0.0000001$. Since the correct answer occurs with probability at least Δ, we have:

$$\begin{aligned}\mathbb{P}(\text{Best-of-}n\text{ outputs correct answer}) &= 1-(1-p(1))^n \\ &\le 1-(1-2\Delta)^{\frac{1}{\Delta}} \\ &\le 0.99.\end{aligned}$$

This confirms that best-of-n fails to produce the correct answer with constant probability. □

A.3 Proof of Proposition 4.2

We first introduce the following result that extends any Transformer to a larger vocabulary, so that it only attends to tokens in its original vocabulary.

Proposition A.1 (Extended Representation to Multiple Token Spaces)**.** *For any $H, L, N_{\max} \in \mathbb{Z}_+$, $\mathcal{V}_1 \cap \mathcal{V}_0 = \emptyset$, there exists a general-purpose Transformer ϕ of type $(O(1), O(\log N_{\max}))$ such that for any Transformers $f = (\theta, \mathrm{pe}, (\mathbf{K}_h^{(l)}, \mathbf{Q}_h^{(l)}, \mathbf{V}_h^{(l)})_{h\in[H], l\in[L]}, \vartheta, \mathcal{V}_1)$ over vocabulary $\mathcal{V}_1$, the Transformer $\widetilde{f} = \phi(f_1)$ satisfies the following property: for any token sequence $v = v_1 \cdots v_n$ such that $n \le N_{\max}$, denote $\{i_1 < \cdots < i_m\} = \{i : v_i \in \mathcal{V}_1\}$, then we have*

$$p_{\widetilde{f}}(\cdot|v) = p_f(\cdot|u),$$

where $u = v_{i_1} \cdots v_{i_m}$.

Proof. Set constants B_v, B_{qk}, B_θ such that for any layer l and head h, it holds that $\left\|(\mathbf{Q}_h^{(l)})^\top \mathbf{K}_h^{(l)}\right\|_2 \le B_{qk}$, $\left\|\mathbf{V}_h^{(l)}\right\|_2 \le B_v$, and $\|\theta(v)\|_2 \le B_\theta$ holds for all $v \in \mathcal{V}$. Let $B = (HB_v)^L B_{qk} B_\theta, C = 4B^2 + \log(1/\epsilon), C_0 = 4C$. By Lemma A.3, there exists $\alpha_1, \ldots, \alpha_{N_{\max}}, \beta_0, \beta_1 \in \mathbb{R}^{d_0}$ and $A_0, A_1, A \in \mathbb{R}^{d_0 \times d_0}$ for $d_0 \le O(\log N_{\max})$ such that

1. For any $i \ge j_1, j_2, j_3$:
$$\begin{aligned}(\alpha_i+\beta_1)^\top A_0(\alpha_{j_1}+\beta_1) = (\alpha_i+\beta_1)^\top A_0(\alpha_{j_2}+\beta_1) &\ge (\alpha_i+\beta_1)^\top A_0(\alpha_{j_1}+\beta_0) + C_0 \\ (\alpha_i+\beta_0)^\top A_0(\alpha_i+\beta_0) &\ge (\alpha_i+\beta_0)^\top A_0(\alpha_{j_1}+\beta_1) + C_0,\end{aligned} \tag{3}$$

2. For any $i > j$
$$\begin{aligned}(\alpha_i+\beta_1)^\top A(\alpha_i+\beta_1) &\ge (\alpha_i+\beta_1)^\top A(\alpha_j+\beta_1) + C_0 \\ &\ge (\alpha_i+\beta_1)^\top A(\alpha_j+\beta_0) + 2C_0,\end{aligned} \tag{4}$$

3. For any $i \ge j, j_1$
$$\begin{aligned}(\alpha_i+\beta_1)^\top A_1(\alpha_j+\beta_0) &= (\alpha_i+\beta_1)^\top A_1(\alpha_{j_1}+\beta_1) + C_0 \\ (\alpha_i+\beta_1)^\top A_1(\alpha_i+\beta_1) &\ge \max\{(\alpha_i+\beta_1)^\top A_1(\alpha_{j_1}+\beta_1), (\alpha_i+\beta_1)^\top A_1(\alpha_{j_1}+\beta_0)\} + C_0.\end{aligned} \tag{5}$$

We define ϕ as follows: for any Transformers $f = (\theta, \mathrm{pe}, (\mathbf{K}_h^{(l)}, \mathbf{Q}_h^{(l)}, \mathbf{V}_h^{(l)})_{h\in[H], l\in[L]}, \vartheta, \mathcal{V}_1)$, the Transformer $\widetilde{f} = \phi(f)$ is given by

$$(\widetilde{\theta}, \widetilde{\mathrm{pe}}, (\widetilde{\mathbf{K}}_h^{(l)}, \widetilde{\mathbf{Q}}_h^{(l)}, \widetilde{\mathbf{V}}_h^{(l)})_{h\in[H+1], l\in[L]}, \widetilde{\vartheta}, \mathcal{V}_1 \cup \mathcal{V}_0),$$

where the tokenizer is given by

$$\widetilde{\theta}(v) = \mathbb{1}(v \in \mathcal{V}_1) \cdot \begin{pmatrix}\theta(v) \\ \beta_1\end{pmatrix} + \mathbb{1}(v \in \mathcal{V}_0) \cdot \begin{pmatrix}0 \\ \beta_0\end{pmatrix},$$

the positional encoder is given by

$$\widetilde{\mathrm{pe}}\left(\begin{pmatrix}x \\ y\end{pmatrix}; v_1, \ldots, v_i\right) = \begin{pmatrix}\mathrm{pe}\,(x; u) \\ \alpha_i + y\end{pmatrix},$$

where $u = v_{i_1} \cdots v_{i_m}$ and $x \in \mathbb{R}^d$; for $l = 1, \ldots, L$ the key, query, value matrices are given by

$$\widetilde{\mathbf{K}}_h^{(l)} = \begin{pmatrix} \mathbf{K}_h^{(l)} & \\ & A_0 \end{pmatrix}, \ \widetilde{\mathbf{Q}}_h^{(l)} = \begin{pmatrix} \mathbf{Q}_h^{(l)} & \\ & I \end{pmatrix},$$

$$\widetilde{\mathbf{V}}_h^{(l)} = \begin{pmatrix} \mathbf{V}_h^{(l)} & \\ & 0 \end{pmatrix},$$

$$\widetilde{\mathbf{K}}_{H+1}^{(l)} = \begin{pmatrix} 0 & \\ & A \end{pmatrix}, \ \widetilde{\mathbf{Q}}_{H+1}^{(l)} = \begin{pmatrix} 0 & \\ & I \end{pmatrix}, \ \widetilde{\mathbf{V}}_{H+1}^{(l)} = \begin{pmatrix} 0 & \\ & I \end{pmatrix}.$$

The output feature is given by $\widetilde{\vartheta}(y) = \begin{pmatrix} \vartheta(y) \\ 0 \end{pmatrix}$. Since $i_1, \ldots, i_m$ only depends on whether v_i's belong to the set $\mathcal{V}_1$, the generalized position encoding pe is well-defined. It can be verified that ϕ is indeed a general-purpose Transformer of type $(O(1), O(\log N_{\max}))$.

We show that for any $l = 1, \ldots, L$,

$$\widetilde{X}_i^{(l)} = \begin{pmatrix} X_i^{(l)} \\ \widetilde{\alpha}_i \end{pmatrix}, \ \forall i = i_1, \ldots, i_m \tag{6}$$

where $X_i^{(l)}$ is the l-th layer of Transformer f at position i (attending only to positions $i_1, \ldots, i_m$) such that

$$\|X_i^{(l)}\|_2 \leq B_\theta (H B_v)^l, \tag{7}$$

and

$$\widetilde{X}_j^{(l)} = \begin{pmatrix} 0 \\ \widetilde{\alpha}_j \end{pmatrix}, \ \forall j \notin \{i_1, \ldots, i_m\} \tag{8}$$

where $\widetilde{\alpha}_i = \alpha_i + \mathbb{1}(v \in \mathcal{V}_0) \cdot \beta_0 + \mathbb{1}(v \in \mathcal{V}_1) \cdot \beta_1$.

We prove these results by induction. The case $l = 1$ folows directly from the definitions of the tokenizer.

Prove Eq. (6). Suppose Eq. (6) and Eq. (8) hold for $1, \ldots, l-1$-th layer, and consider l-the layer. We have

$$\widetilde{X}_i^{(l+1)} = \underbrace{\sum_{h=1}^{H} \sum_{j=1}^{i} \frac{\exp\left((\widetilde{\mathbf{Q}}_h^{(l)} \widetilde{X}_i^{(l)})^\top (\widetilde{\mathbf{K}}_h^{(l)} \widetilde{X}_j^{(l)}) \right)}{\widetilde{Z}_h^{(l)}} \cdot \widetilde{\mathbf{V}}_h^{(l)} \widetilde{X}_j^{(l)}}_{\text{term 1}} + \underbrace{\sum_{j=1}^{i} \frac{\exp\left((\widetilde{\mathbf{Q}}_{H+1}^{(l)} \widetilde{X}_i^{(l)})^\top (\widetilde{\mathbf{K}}_{H+1}^{(l)} \widetilde{X}_j^{(l)}) \right)}{\widetilde{Z}_{H+1}^{(l)}} \cdot \widetilde{\mathbf{V}}_{H+1}^{(l)} \widetilde{X}_j^{(l)}}_{\text{term 2}}.$$

Eq. (3) ensures that for any $i, i' \in \{i_1, \ldots, i_m\}, j \notin \{i_1, \ldots, i_m\}$:

$$\begin{aligned} (\widetilde{\mathbf{Q}}_h^{(l)} \widetilde{X}_i^{(l)})^\top (\widetilde{\mathbf{K}}_h^{(l)} \widetilde{X}_{i'}^{(l)}) &= (\mathbf{Q}_h^{(l)} \widetilde{X}_i^{(l)})^\top (\mathbf{K}_h^{(l)} \widetilde{X}_{i'}^{(l)}) + (\alpha_i + \beta_1)^\top A_0 (\alpha_{i'} + \beta_1) \\ &\geq (\mathbf{Q}_h^{(l)} X_i^{(l)})^\top (\mathbf{K}_h^{(l)} X_j^{(l)}) + (\alpha_i + \beta_1)^\top A_0 (\alpha_j + \beta_0) + C \\ &= (\widetilde{\mathbf{Q}}_h^{(l)} \widetilde{X}_i^{(l)})^\top (\widetilde{\mathbf{K}}_h^{(l)} \widetilde{X}_j^{(l)}) + C, \end{aligned}$$

and if $i, j_1, j_2 \in \{i_1, \ldots, i_m\}$

$$\begin{aligned} &(\widetilde{\mathbf{Q}}_h^{(l)} \widetilde{X}_i^{(l)})^\top (\widetilde{\mathbf{K}}_h^{(l)} \widetilde{X}_{j_1}^{(l)}) - (\widetilde{\mathbf{Q}}_h^{(l)} \widetilde{X}_i^{(l)})^\top (\widetilde{\mathbf{K}}_h^{(l)} \widetilde{X}_{j_2}^{(l)}) \\ &= (\mathbf{Q}_h^{(l)} X_i^{(l)})^\top (\mathbf{K}_h^{(l)} X_{j_1}^{(l)}) + (\alpha_i + \beta_1)^\top A_0 (\alpha_{j_1} + \beta_1)) - (\mathbf{Q}_h^{(l)} X_i^{(l)})^\top (\mathbf{K}_h^{(l)} X_{j_2}^{(l)}) - (\alpha_i + \beta_1)^\top A_0 (\alpha_{j_2} + \beta_1) \\ &= (\mathbf{Q}_h^{(l)} X_i^{(l)})^\top (\mathbf{K}_h^{(l)} \widetilde{X}_{j_1}^{(l)}) - (\mathbf{Q}_h^{(l)} \widetilde{X}_i^{(l)})^\top (\mathbf{K}_h^{(l)} X_{j_2}^{(l)}), \end{aligned}$$

where we use the fact that $C_0 \geq C + 2 \max_{h,l,i,j} \left| (\mathbf{Q}_h^{(l)} X_i^{(l)})^\top (\mathbf{K}_h^{(l)} X_j^{(l)}) \right|$. Since the transformers have precision ϵ and $C \geq 2 \max_{h,l,i,j} \left| (\mathbf{Q}_h^{(l)} X_i^{(l)})^\top (\mathbf{K}_h^{(l)} X_j^{(l)}) \right| + \log(1/\epsilon)$, it follows that the attention weights of head $(k-1)H+h$ is identical to the attention weights of expert k, i.e.

$$\frac{\exp\left((\widetilde{\mathbf{Q}}_h^{(l)}\widetilde{X}_i^{(l)})^\top(\widetilde{\mathbf{K}}_h^{(l)}\widetilde{X}_j^{(l)})\right)}{\widetilde{Z}_h^{(l)}} = \mathbb{1}(j\in\{i_1,\ldots,i_m\})\cdot\frac{\exp\left((\mathbf{Q}_h^{(l)}X_i^{(l)})^\top(\mathbf{K}_h^{(l)}X_j^{(l)})\right)}{Z_h^{(l)}}.$$

Therefore

$$\text{term 1} = \sum_{h=1}^{H}\sum_{j=i_1,\ldots,i_m}\frac{\exp\left((\mathbf{Q}_h^{(l)}X_i^{(l)})^\top(\mathbf{K}_h^{(l)}X_j^{(l)})\right)}{Z_h^{(l)}}\cdot\begin{pmatrix}\mathbf{V}_h^{(l)}X_j^{(l)}\\ 0\end{pmatrix} = \begin{pmatrix}X_j^{(l+1)}\\ 0\end{pmatrix}.$$

Furthermore, by Eq. (4) we have for any $j<i$

$$\begin{aligned}(\widetilde{\mathbf{Q}}_{H+1}^{(l)}\widetilde{X}_i^{(l)})^\top(\widetilde{\mathbf{K}}_{H+1}^{(l)}\widetilde{X}_i^{(l)}) &= \widetilde{\alpha}_i^\top A\widetilde{\alpha}_i\\ &\geq \widetilde{\alpha}_i^\top A\widetilde{\alpha}_j + C\\ &= (\widetilde{\mathbf{Q}}_{H+1}^{(l)}\widetilde{X}_i^{(l)})^\top(\widetilde{\mathbf{K}}_{H+1}^{(l)}\widetilde{X}_j^{(l)}) + C,\end{aligned}$$

and hence the attention weighs concentrates on i itself. Thus

$$\text{term 2} = \begin{pmatrix}0 & \\ & I\end{pmatrix}\cdot\begin{pmatrix}X_i^{(l)}\\ \widetilde{\alpha}_i\end{pmatrix} = \begin{pmatrix}0\\ \widetilde{\alpha}_i\end{pmatrix}.$$

Combining, we derive Eq.(6) for $(l+1)$-th layer.

Prove Eq. (7). From above,

$$\begin{aligned}\|X_i^{(l+1)}\|_2 &= \left\|\sum_{h=1}^{H}\sum_{j=1}^{i}\frac{\exp\left((\widetilde{\mathbf{Q}}_h^{(l)}\widetilde{X}_i^{(l)})^\top(\widetilde{\mathbf{K}}_h^{(l)}\widetilde{X}_j^{(l)})\right)}{\widetilde{Z}_h^{(l)}}\cdot\mathbf{V}_h^{(l)}X_j^{(l)}\right\|_2\\ &\leq HB_v\cdot\max_{j\leq i}\|X_j^{(l)}\|_2\\ &\leq B_\theta(HB_v)^{l+1}.\end{aligned}$$

This confirms Eq. (24) for $l+1$.

Prove Eq. (8). Notice that Eq. (3) ensures that for any $j,j'\notin\{i: v_i\in\mathcal{V}_1\}$ and $i\in\{i: v_i\in\mathcal{V}_1\}$:

$$\begin{aligned}(\widetilde{\mathbf{Q}}_h^{(l)}\widetilde{X}_j^{(l)})^\top(\widetilde{\mathbf{K}}_h^{(l)}\widetilde{X}_{j'}^{(l)}) &= (\mathbf{Q}_h^{(l)}X_j^{(l)})^\top(\mathbf{K}_h^{(l)}X_{j'}^{(l)}) + (\alpha_j+\beta_0)^\top A_0(\alpha_{j'}+\beta_0)\\ &\geq (\mathbf{Q}_h^{(l)}X_j^{(l)})^\top(\mathbf{K}_h^{(l)}X_i^{(l)}) + (\alpha_j+\beta_0)^\top A_0(\alpha_i+\beta_1) + C\\ &= (\widetilde{\mathbf{Q}}_h^{(l)}\widetilde{X}_j^{(l)})^\top(\widetilde{\mathbf{K}}_h^{(l)}\widetilde{X}_i^{(l)}) + C.\end{aligned}$$

It follows that the attention weights is concentrated on the compliment of $\{i: v_i\in\mathcal{V}_1\}$ itself, and therefore Eq. (8) follows by a simple induction argument.

Finally, at the output layer

$$\begin{aligned}p_{\widetilde{f}}(y|v_1,\ldots,v_n) &= \mathrm{Softmax}(\widetilde{\vartheta}(y)^\top\widetilde{X}_n^{(L)})\\ &= \mathrm{Softmax}(\vartheta(y)^\top X_m^{(L)})\\ &= p_{f_\kappa}(y|u).\end{aligned}$$

This establishes the desired statement. □

Now we return to the proof of Proposition 4.2.

Proof. By Proposition A.1, it suffices to construct general-purpose Transformer ϕ such that

$$p_{\widetilde{f}}(\cdot|v) = p_{f_\kappa}(\cdot|u),$$

where $u = v_1\cdots v_{i_0-1}v_{i_0+1}\cdots v_n$, because then the $\widetilde{\phi}$ given by

$$\widetilde{\phi}(f_1,\ldots,f_K) = \phi(\phi_e(f_1),\ldots,\phi_e(f_K))$$

satisfies the requirement, where ϕ_e is the general-purpose Transformer that extends the K Transformers to the larger vocabulary $\mathcal{V} := \cup_{k=1}^K\mathcal{V}_k$ as given by Proposition A.1.

Set constants B_v, B_{qk}, B_θ such that for any layer l and head h, it holds that $\left\|(\mathbf{Q}_h^{(l)})^\top \mathbf{K}_h^{(l)}\right\|_2 \leq B_{qk}$, $\left\|\mathbf{V}_h^{(l)}\right\|_2 \leq B_v$, and $\|\theta(v)\|_2 \leq B_\theta$ holds for all $v \in \mathcal{V}$. Let $B = (KHB_v)^L B_{qk} B_\theta, C = 4B^2 + \log(1/\epsilon), C_0 = 4C$. By Lemma A.3, there exists $\alpha_1, \dots, \alpha_N, \beta_0, \beta_1, \dots, \beta_K \in \mathbb{R}^{d_0}$ and $A_1, \dots, A_K \in \mathbb{R}^{d_0 \times d_0}$ for $d_0 \leq O(K + \log N_{\max})$ such that

1. For any $i \geq j_1, j_2, j_3$ and $k, k', k'' \neq 0$:
$$(\alpha_i + \beta_k)^\top A_0(\alpha_{j_1} + \beta_{k'}) = (\alpha_i + \beta_k)^\top A_0(\alpha_{j_2} + \beta_{k''}) \geq (\alpha_i + \beta_k)^\top A_0(\alpha_{j_1} + \beta_0) + C_0$$
$$(\alpha_i + \beta_0)^\top A_0(\alpha_i + \beta_0) \geq (\alpha_i + \beta_0)^\top A_0(\alpha_{j_1} + \beta_k) + C_0, \tag{9}$$
2. For any $i > j$ and $k \neq k' \neq 0$
$$\begin{aligned}(\alpha_i + \beta_k)^\top A(\alpha_i + \beta_k) &\geq (\alpha_i + \beta_k)^\top A(\alpha_j + \beta_{k'}) + C_0 \\ &\geq (\alpha_i + \beta_k)^\top A(\alpha_j + \beta_0) + 2C_0,\end{aligned} \tag{10}$$
3. For any $i \geq j, j_1$ and $k \neq k', k''$
$$(\alpha_i + \beta_k)^\top A_{k'}(\alpha_j + \beta_0) = (\alpha_i + \beta_k)^\top A_{k'}(\alpha_{j_1} + \beta_{k''}) + C_0$$
$$(\alpha_i + \beta_k)^\top A_k(\alpha_i + \beta_k) \geq \max\{(\alpha_i + \beta_k)^\top A_k(\alpha_{j_1} + \beta_{k''}), (\alpha_i + \beta_k)^\top A_{k'}(\alpha_{j_1} + \beta_0)\} + C_0, \tag{11}$$

We define ϕ as follows: for any Transformers
$$f_k = (\theta_k, \mathrm{pe}_k, (\mathbf{K}_{k;h}^{(l)}, \mathbf{Q}_{k;h}^{(l)}, \mathbf{V}_{k;h}^{(l)})_{h\in[H], l\in[L]}, \vartheta_k, \mathcal{V}_k),$$
over $\mathcal{V}_k,\ k \in [K]$, the Transformer $\widetilde{f} = \phi(f_1, \dots, f_K)$ is given by
$$(\widetilde{\theta}, \widetilde{\mathrm{pe}}, (\widetilde{\mathbf{K}}_h^{(l)}, \widetilde{\mathbf{Q}}_h^{(l)}, \widetilde{\mathbf{V}}_h^{(l)})_{h\in[KH+1], l\in[L+1]}, \widetilde{\vartheta}, \mathcal{V}),$$
where the tokenizer is given by
$$\widetilde{\theta}(v) = \mathbb{1}(v \notin \mathcal{V}_0) \cdot \begin{pmatrix} \theta_1(v) \\ \vdots \\ \theta_K(v) \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \beta_{\mathcal{E}(v)} \end{pmatrix}$$
where $\mathcal{E}(v) = k$ iff $v \in \mathcal{V}_k$. Let the positional encoder be given by
$$\widetilde{\mathrm{pe}}\left(\begin{pmatrix} x \\ y \end{pmatrix}; v_1, \dots, v_i\right) = \begin{pmatrix} \mathrm{pe}_1(x; u) \\ \vdots \\ \mathrm{pe}_K(x; u) \\ \alpha_i + y \end{pmatrix},$$
where $x \in \mathbb{R}^d$ and u is the sub-sequence of v that omits v_{i_0} (if any); for $l = 1, \dots, L$ the key, query, value matrices are given by
$$\widetilde{\mathbf{K}}_{(k-1)H+h}^{(l)} = \begin{pmatrix} 0 & & & \\ & \ddots & & \\ & & \mathbf{K}_{k;h}^{(l)} & & \\ & & & \ddots & \\ & & & & A_0 \end{pmatrix}, \ \widetilde{\mathbf{Q}}_{(k-1)H+h}^{(l)} = \begin{pmatrix} 0 & & & \\ & \ddots & & \\ & & \mathbf{Q}_{k;h}^{(l)} & & \\ & & & \ddots & \\ & & & & I \end{pmatrix},$$
$$\widetilde{\mathbf{V}}_{(k-1)H+h}^{(l)} = \begin{pmatrix} 0 & & & \\ & \ddots & & \\ & & \mathbf{V}_{k;h}^{(l)} & & \\ & & & \ddots & \\ & & & & 0 \end{pmatrix},$$

$$\widetilde{\mathbf{K}}_{KH+1}^{(l)} = \begin{pmatrix} 0 & & & \\ & \ddots & & \\ & & 0 & \\ & & & A \end{pmatrix}, \ \widetilde{\mathbf{Q}}_{KH+1}^{(l)} = \begin{pmatrix} 0 & & & \\ & \ddots & & \\ & & 0 & \\ & & & I \end{pmatrix}, \ \widetilde{\mathbf{V}}_{KH+1}^{(l)} = \begin{pmatrix} 0 & & & \\ & \ddots & & \\ & & 0 & \\ & & & I \end{pmatrix},$$

where the submatrices $\mathbf{K}_{k;h}^{(l)}, \mathbf{Q}_{k;h}^{(l)}, \mathbf{V}_{k;h}^{(l)}$ are located in the k-th diagonal block, and for the final layer

$$\widetilde{\mathbf{K}}_{k}^{(L+1)} = \begin{pmatrix} 0 & & & \\ & \ddots & & \\ & & 0 & \\ & & & A_k \end{pmatrix}, \ \widetilde{\mathbf{Q}}_{k}^{(L+1)} = \begin{pmatrix} 0 & & & \\ & \ddots & & \\ & & 0 & \\ & & & I \end{pmatrix}, \ \widetilde{\mathbf{V}}_{k}^{(L+1)} = \begin{pmatrix} 0 & & & & \\ & \ddots & & & \\ & & I & & \\ & & & \ddots & \\ & & & & 0 \end{pmatrix},$$

where the identity sub-matrix in $\widetilde{\mathbf{V}}_{k}^{(L+1)}$ is located in the k-th block. The output feature is given by $\widetilde{\vartheta}(y) = \begin{pmatrix} \vartheta_1(y) \\ \vdots \\ \vartheta_K(y) \\ 0 \end{pmatrix}$. Since $u^{(k)}$'s only depend on set membership information of v_i's, the generalized position encoding pe is well-defined. We can easily verify that ϕ is indeed a general-purpose Transformer of type $(O(K), O(\log N_{\max}))$.

We show that for any $l = 1, \ldots, L$,

$$\widetilde{X}_i^{(l)} = \begin{pmatrix} X_{1;i}^{(l)} \\ \vdots \\ X_{K;i}^{(l)} \\ \widetilde{\alpha}_i \end{pmatrix}, \ \forall i \neq i_0 \tag{12}$$

where $X_{k;i}^{(l)}$ is the l-th layer of Transformer k at position i (attending to all positions but i_0) such that

$$\|X_{k;i}^{(l)}\|_2 \leq B_\theta (KHB_v)^l. \tag{13}$$

and

$$\widetilde{X}_{i_0}^{(l)} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \widetilde{\alpha}_{i_0} \end{pmatrix} \tag{14}$$

where $\widetilde{\alpha}_i = \alpha_i + \beta_{\mathcal{E}(v_i)}$.

We prove these results by induction. The case $l = 1$ folows directly from the definitions of the tokenizer.

Prove Eq. (12). Suppose Eq. (12) and Eq. (14) hold for $1, \ldots, l-1$=th layer, and consider l-the layer. We have

$$\widetilde{X}_i^{(l+1)} = \underbrace{\sum_{k=1}^{K}\sum_{h=1}^{H}\sum_{j=1}^{i} \frac{\exp\left((\widetilde{\mathbf{Q}}_{(k-1)H+h}^{(l)}\widetilde{X}_i^{(l)})^\top(\widetilde{\mathbf{K}}_{(k-1)H+h}^{(l)}\widetilde{X}_j^{(l)})\right)}{\widetilde{Z}_{(k-1)H+h}^{(l)}} \cdot \widetilde{\mathbf{V}}_{(k-1)H+h}^{(l)}\widetilde{X}_j^{(l)}}_{\text{term 1}}$$
$$+ \underbrace{\sum_{j=1}^{i} \frac{\exp\left((\widetilde{\mathbf{Q}}_{KH+1}^{(l)}\widetilde{X}_i^{(l)})^\top(\widetilde{\mathbf{K}}_{KH+1}^{(l)}\widetilde{X}_j^{(l)})\right)}{\widetilde{Z}_{KH+1}^{(l)}} \cdot \widetilde{\mathbf{V}}_{KH+1}^{(l)}\widetilde{X}_j^{(l)}}_{\text{term 2}}.$$

Eq. (9) ensures that for any $j_1 < j_2 \leq i$ such that $i_0 \notin \{i, j_1, j_2\}$:

$$\begin{aligned}
(\widetilde{\mathbf{Q}}^{(l)}_{(k-1)H+h}\widetilde{X}^{(l)}_i)^\top(\widetilde{\mathbf{K}}^{(l)}_{(k-1)H+h}\widetilde{X}^{(l)}_{j_1}) &= (\mathbf{Q}^{(l)}_{k;h}X^{(l)}_{k;i})^\top(\mathbf{K}^{(l)}_{k;h}X^{(l)}_{k;j_1}) + (\alpha_i + \beta_{\mathcal{E}(i)})^\top A_0(\alpha_{j_1} + \beta_{\mathcal{E}(j_1)}) \\
&\geq (\mathbf{Q}^{(l)}_{k;h}X^{(l)}_{k;i})^\top(\mathbf{K}^{(l)}_{k;h}X^{(l)}_{k;j_1}) + (\alpha_i + \beta_{\mathcal{E}(i)})^\top A_0(\alpha_{i_0} + \beta_{\mathcal{E}(i_0)}) + C \\
&= (\widetilde{\mathbf{Q}}^{(l)}_{(k-1)H+h}\widetilde{X}^{(l)}_i)^\top(\widetilde{\mathbf{K}}^{(l)}_{(k-1)H+h}\widetilde{X}^{(l)}_{i_0}) + C.
\end{aligned}$$

and

$$\begin{aligned}
&(\widetilde{\mathbf{Q}}^{(l)}_{(k-1)H+h}\widetilde{X}^{(l)}_i)^\top(\widetilde{\mathbf{K}}^{(l)}_{(k-1)H+h}\widetilde{X}^{(l)}_{j_1}) - (\widetilde{\mathbf{Q}}^{(l)}_{(k-1)H+h}\widetilde{X}^{(l)}_i)^\top(\widetilde{\mathbf{K}}^{(l)}_{(k-1)H+h}\widetilde{X}^{(l)}_{j_2}) \\
={}& (\mathbf{Q}^{(l)}_{k;h}X^{(l)}_{k;i})^\top(\mathbf{K}^{(l)}_{k;h}X^{(l)}_{k;j_1}) + (\alpha_i + \beta_{\mathcal{E}(i)})^\top A_0(\alpha_{j_1} + \beta_{\mathcal{E}(j_1)}) \\
&- (\mathbf{Q}^{(l)}_{k;h}X^{(l)}_{k;i})^\top(\mathbf{K}^{(l)}_{k;h}X^{(l)}_{k;j_2}) - (\alpha_i + \beta_{\mathcal{E}(i)})^\top A_0(\alpha_{j_2} + \beta_{\mathcal{E}(j_2)}) \\
={}& (\mathbf{Q}^{(l)}_{k;h}X^{(l)}_{k;i})^\top(\mathbf{K}^{(l)}_{k;h}X^{(l)}_{k;j_1}) - (\mathbf{Q}^{(l)}_{k;h}X^{(l)}_{k;i})^\top(\mathbf{K}^{(l)}_{k;h}X^{(l)}_{k;j_2}).
\end{aligned}$$

It follows from the precision ϵ of the transformers that the attention weights of head $(k-1)H + h$ is identical to the attention weights of expert k, i.e.

$$\frac{\exp\left((\widetilde{\mathbf{Q}}^{(l)}_{(k-1)H+h}\widetilde{X}^{(l)}_i)^\top(\widetilde{\mathbf{K}}^{(l)}_{(k-1)H+h}\widetilde{X}^{(l)}_j)\right)}{\widetilde{Z}^{(l)}_{(k-1)H+h}} = \frac{\exp\left((\mathbf{Q}^{(l)}_{k;h}X^{(l)}_{k;i})^\top(\mathbf{K}^{(l)}_{k;h}X^{(l)}_{k;j})\right)}{Z^{(l)}_{k;h}}.$$

Therefore

$$\text{term 1} = \sum_{k=1}^{K}\sum_{h=1}^{H}\sum_{j=1}^{i}\frac{\exp\left((\mathbf{Q}^{(l)}_{k;h}X^{(l)}_{k;i})^\top(\mathbf{K}^{(l)}_{k;h}X^{(l)}_{k;j})\right)}{Z^{(l)}_{k;h}} \cdot \begin{pmatrix} 0 \\ \vdots \\ \mathbf{V}^{(l)}_{k;h}X^{(l)}_{k;j} \\ \vdots \\ 0 \end{pmatrix} = \begin{pmatrix} X^{(l)}_{1;i} \\ \vdots \\ X^{(l)}_{K;i} \\ 0 \end{pmatrix}.$$

Furthermore, by Eq. (10) we have for any $j < i$

$$\begin{aligned}
(\widetilde{\mathbf{Q}}^{(l)}_{KH+1}\widetilde{X}^{(l)}_i)^\top(\widetilde{\mathbf{K}}^{(l)}_{KH+1}\widetilde{X}^{(l)}_i) &= \widetilde{\alpha}_i^\top A\widetilde{\alpha}_i \\
&\geq \widetilde{\alpha}_i^\top A\widetilde{\alpha}_j + C \\
&= (\widetilde{\mathbf{Q}}^{(l)}_{KH+1}\widetilde{X}^{(l)}_i)^\top(\widetilde{\mathbf{K}}^{(l)}_{KH+1}\widetilde{X}^{(l)}_j) + C
\end{aligned}$$

and hence the attention weighs concentrates on i itself. Thus

$$\text{term 2} = \begin{pmatrix} 0 & & & \\ & \ddots & & \\ & & 0 & \\ & & & I \end{pmatrix} \cdot \begin{pmatrix} X^{(l)}_{1;i} \\ \vdots \\ X^{(l)}_{K;i} \\ \widetilde{\alpha}_i \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \widetilde{\alpha}_i \end{pmatrix}$$

Combining these two terms, we confirm that Eq.(12) holds for $(l+1)$-th layer.

Prove Eq. (13)**.** From above,

$$\begin{aligned}
\|X^{(l+1)}_{k;i}\|_2 &= \left\|\sum_{k=1}^{K}\sum_{h=1}^{H}\sum_{j=1}^{i}\frac{\exp\left((\widetilde{\mathbf{Q}}^{(l)}_{(k-1)H+h}\widetilde{X}^{(l)}_i)^\top(\widetilde{\mathbf{K}}^{(l)}_{(k-1)H+h}\widetilde{X}^{(l)}_j)\right)}{\widetilde{Z}^{(l)}_{(k-1)H+h}} \cdot \mathbf{V}^{(l)}_{k;h}X^{(l)}_{k;j}\right\|_2 \\
&\leq KHB_v \cdot \max_{j \leq i}\|X^{(l)}_{k;j}\|_2 \\
&\leq B_\theta(KHB_v)^{l+1}.
\end{aligned}$$

This confirms Eq. (13) for $l+1$.

Prove Eq. (14)**.** Notice that Eq. (9) ensures that for any $j \le i_0$:

$$\begin{aligned}(\widetilde{\mathbf{Q}}^{(l)}_{(k-1)H+h}\widetilde{X}^{(l)}_{i_0})^\top(\widetilde{\mathbf{K}}^{(l)}_{(k-1)H+h}\widetilde{X}^{(l)}_{i_0}) &= (\mathbf{Q}^{(l)}_{k;h}X^{(l)}_{k;i_0})^\top(\mathbf{K}^{(l)}_{k;h}X^{(l)}_{k;i_0}) + (\alpha_{i_0}+\beta_{\mathcal{E}(i_0)})^\top A_0(\alpha_{i_0}+\beta_{\mathcal{E}(i_0)})\\ &\ge (\mathbf{Q}^{(l)}_{k;h}X^{(l)}_{k;i_0})^\top(\mathbf{K}^{(l)}_{k;h}X^{(l)}_{k;j}) + (\alpha_{i_0}+\beta_{\mathcal{E}(i_0)})^\top A_0(\alpha_j+\beta_{\mathcal{E}(j)}) + C\\ &= (\widetilde{\mathbf{Q}}^{(l)}_{(k-1)H+h}\widetilde{X}^{(l)}_{i_0})^\top(\widetilde{\mathbf{K}}^{(l)}_{(k-1)H+h}\widetilde{X}^{(l)}_{j}) + C.\end{aligned}$$

It follows that the attention weights of head $(k-1)H+h$ is concentrated on i_0 itself, therefore

$$\text{term 1} = \sum_{k=1}^{K}\sum_{h=1}^{H}\begin{pmatrix}0\\ \vdots\\ \mathbf{V}^{(l)}_{k;h}\cdot 0\\ \vdots\\ 0\end{pmatrix} = 0.$$

By the same argument, for $i = i_0$ we have

$$\text{term 2} = \begin{pmatrix}0 & & & \\ & \ddots & & \\ & & 0 & \\ & & & I\end{pmatrix}\cdot\begin{pmatrix}0\\ \vdots\\ 0\\ \widetilde{\alpha}_{i_0}\end{pmatrix} = \begin{pmatrix}0\\ \vdots\\ 0\\ \widetilde{\alpha}_{i_0}\end{pmatrix}.$$

Combining these confirms Eq. (14).

Next, we show that the last layer satisfies

$$\widetilde{X}^{(L+1)}_n = \begin{pmatrix}0\\ \vdots\\ X^{(L+1)}_{\kappa;n}\\ \vdots\\ 0\end{pmatrix} \tag{15}$$

where $X^{(L+1)}_{\kappa;n}$ is the κ-th block. To see this, we notice that Eq. (11) implies the followings (the proofs are identical to the above):

1. Attention sink to dummny token v_{i_0} for mismatch expert: for any $k' \ne \kappa$ and $j \le n$ we have
$$\begin{aligned}(\widetilde{\mathbf{Q}}^{(L)}_{(k'-1)H+h}\widetilde{X}^{(L)}_{n})^\top(\widetilde{\mathbf{K}}^{(L)}_{(k'-1)H+h}\widetilde{X}^{(L)}_{j}) &= (\alpha_n+\beta_{\mathcal{E}(n)})^\top A_{k'}(\alpha_j+\beta_{\mathcal{E}(j)})\\ &\le (\alpha_n+\beta_{\mathcal{E}(n)})^\top A_{k'}(\alpha_{i_0}+\beta_{\mathcal{E}(i_0)}) - C\\ &= (\widetilde{\mathbf{Q}}^{(L)}_{(k'-1)H+h}\widetilde{X}^{(L)}_{n})^\top(\widetilde{\mathbf{K}}^{(L)}_{(k'-1)H+h}\widetilde{X}^{(L)}_{i_0}) - C.\end{aligned} \tag{16}$$

2. Attention to oneself for matching expert: for any $j \ne i_0$ we have
$$\begin{aligned}(\widetilde{\mathbf{Q}}^{(L)}_{(\kappa-1)H+h}\widetilde{X}^{(L)}_{n})^\top(\widetilde{\mathbf{K}}^{(L)}_{(\kappa-1)H+h}\widetilde{X}^{(L)}_{j}) &= (\alpha_n+\beta_{\mathcal{E}(n)})^\top A_{\kappa}(\alpha_j+\beta_{\mathcal{E}(j)})\\ &\ge (\alpha_n+\beta_{\mathcal{E}(n)})^\top A_{\kappa}(\alpha_{i_0}+\beta_{\mathcal{E}(i_0)}) + C\\ &= (\widetilde{\mathbf{Q}}^{(L)}_{(\kappa-1)H+h}\widetilde{X}^{(L)}_{n})^\top(\widetilde{\mathbf{K}}^{(L)}_{(\kappa-1)H+h}\widetilde{X}^{(L)}_{i_0}) + C,\end{aligned} \tag{17}$$
and
$$\begin{aligned}(\widetilde{\mathbf{Q}}^{(L)}_{(\kappa-1)H+h}\widetilde{X}^{(L)}_{n})^\top(\widetilde{\mathbf{K}}^{(L)}_{(\kappa-1)H+h}\widetilde{X}^{(L)}_{n}) &= (\alpha_n+\beta_{\mathcal{E}(n)})^\top A_{\kappa}(\alpha_n+\beta_{\mathcal{E}(n)})\\ &\ge (\alpha_n+\beta_{\mathcal{E}(n)})^\top A_{\kappa}(\alpha_j+\beta_{\mathcal{E}(j)}) + C\\ &= (\widetilde{\mathbf{Q}}^{(L)}_{(\kappa-1)H+h}\widetilde{X}^{(L)}_{n})^\top(\widetilde{\mathbf{K}}^{(L)}_{(\kappa-1)H+h}\widetilde{X}^{(L)}_{j}) + C.\end{aligned} \tag{18}$$

Combining Eq. (16), Eq. (17), and Eq. (18), we have

$$\frac{\exp\left((\widetilde{\mathbf{Q}}^{(L)}_{(k-1)H+h}\widetilde{X}^{(L)}_n)^\top(\widetilde{\mathbf{K}}^{(L)}_{(k-1)H+h}\widetilde{X}^{(L)}_j)\right)}{Z^{(l)}_k} = \begin{cases}\delta^{i_0}_j, & k \neq \kappa \\ \delta^n_j, & k = \kappa\end{cases}$$

It follows that

$$\begin{aligned}\widetilde{X}^{(L+1)}_n &= \widetilde{\mathbf{V}}^{(L)}_{(\kappa-1)H+h}\cdot\widetilde{X}^{(L)}_n + \sum_{k\neq\kappa}\mathbf{V}^{(L)}_{(\kappa-1)H+h}\cdot\widetilde{X}^{(L)}_{i_0} \\ &= \begin{pmatrix}0 & & & \\ & \ddots & & \\ & & I & \\ & & & \ddots & \\ & & & & 0\end{pmatrix}\cdot\begin{pmatrix}X^{(L)}_{1;i}\\ \vdots \\ X^{(L)}_{K;i}\\ \widetilde{\alpha}_i\end{pmatrix} = \begin{pmatrix}0\\ \vdots\\ X^{(L)}_{\kappa;n}\\ \vdots\\ 0\end{pmatrix}.\end{aligned}$$

Therefore we establish Eq. (15).

Finally, at the output layer

$$\begin{aligned}p_{\widetilde{f}}(y|v_1,\dots,v_n) &= \mathrm{Softmax}(\widetilde{\vartheta}(y)^\top\widetilde{X}^{(L+1)}_n)\\ &= \mathrm{Softmax}(\vartheta(y)^\top Y^{(L)}_{n-1})\\ &= p_{f_\kappa}(y|u).\end{aligned}$$

This establishes the desired statement. □

A.4 Proof of Proposition 4.4

Proof. Set constants B_v, B_{qk}, B_θ such that for any layer l and head h, it holds that $\left\|(\mathbf{Q}^{(l)}_h)^\top\mathbf{K}^{(l)}_h\right\|_2 \leq B_{qk}$, $\left\|\mathbf{V}^{(l)}_h\right\|_2 \leq B_v$, and $\|\theta(v)\|_2 \leq B_\theta$ holds for all $v \in \mathcal{V}$. Let $B = (KHB_v)^L B_{qk}B_\theta, C = 2B^2 + \log(1/\epsilon), C_0 = 4C$. Define $\iota(i) = u$ iff $\xi_u \leq i < \xi_{u+1}$ ($\xi_0 = -1, \xi_{m+1} = \infty$ by default). Let $\mathcal{E}(\cdot)$ denote the task id indicated by the special token. By Lemma A.2, there exists $\alpha_1,\dots,\alpha_N,\beta_1,\dots,\beta_K \in \mathbb{R}^{d_0}$ and $A_1,\dots,A_K \in \mathbb{R}^{d_0\times d_0}$ for $d_0 \leq O(K + \log N_{\max})$ such that for any $n \leq N$ we have

1. For any $k \neq k'$:

$$\alpha_n^\top A_k(\alpha_n + \beta_{k'}) \geq C_0 + \begin{cases}\alpha_n^\top A_k\alpha_n\\ \alpha_n^\top A_k\alpha_j\\ \alpha_n^\top A_k(\alpha_j+\beta_{k''})\end{cases}, \ \forall 0 \leq j \leq n, 1 \leq k'' \leq K. \tag{19}$$

2. For any $k \in [K]$:

$$\alpha_n^\top A_k\alpha_n = \alpha_n^\top A_k\alpha_0 \geq C_0 + \begin{cases}\alpha_n^\top A_k(\alpha_n+\beta_k)\\ \alpha_n^\top A_k\alpha_j\\ \alpha_n^\top A_k(\alpha_j+\beta_{k'})\end{cases}, \ \forall 0 < j < n, k' \neq k. \tag{20}$$

3. For any $k, k', k'' \in [K]$:

$$(\alpha_n + \beta_{k'})^\top A_k(\alpha_n + \beta_{k'}) \geq C_0 + (\alpha_n + \beta_{k'})^\top A_k\alpha_j, \ \forall 0 \leq j \leq n. \tag{21}$$

4. For any $0 < j < n$:

$$\begin{aligned}\alpha_n^\top A\alpha_n &\geq \alpha_n^\top A(\alpha_n + \beta_k) + C_0\\ &\geq C_0 + \max\{\alpha_n^\top A\alpha_j, \alpha_n^\top A(\alpha_j + \beta_{k'})\}, \ \forall k, k'' \in [K].\end{aligned} \tag{22}$$

We define ϕ as follows: for any Transformers

$$f_k = (\theta_k, \mathrm{pe}_k, (\mathbf{K}^{(l)}_{k;h}, \mathbf{Q}^{(l)}_{k;h}, \mathbf{V}^{(l)}_{k;h})_{h\in[H],l\in[L]}, \vartheta_k, \mathcal{V}), k \in [K]$$

over $\mathcal{V}$, the Transformer $\widetilde{f} = \phi(f_1, \ldots, f_K)$ is given by

$$(\widetilde{\theta}, \widetilde{\mathrm{pe}}, (\widetilde{\mathbf{K}}_h^{(l)}, \widetilde{\mathbf{Q}}_h^{(l)}, \widetilde{\mathbf{V}}_h^{(l)})_{h\in[KH+1], l\in[L]}, \widetilde{\vartheta}, \mathcal{V}\cup\Omega),$$

where the tokenizer is given by

$$\widetilde{\theta}(v) = \begin{pmatrix} \theta_1(v) \\ \vdots \\ \theta_K(v) \\ 0 \end{pmatrix},\ v \in \mathcal{V},\ \widetilde{\theta}(\omega) = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \beta_{\mathcal{E}(\omega)} \end{pmatrix},\ \omega \in \Omega,$$

the positional encoder is given by

$$\widetilde{\mathrm{pe}}\left(\begin{pmatrix} x \\ y \end{pmatrix}; v_1, \ldots, v_i \right) = \begin{pmatrix} \mathrm{pe}_1\left(x; v_1, \cdots, v_{\xi_1 - 1}, v_{\xi_m+1}, \cdots, v_n\right) \\ \vdots \\ \mathrm{pe}_K\left(x; v_1, \cdots, v_{\xi_1 - 1}, v_{\xi_m+1}, \cdots, v_n\right) \\ \alpha_{\iota(i)} + y \end{pmatrix},$$

where $x \in \mathbb{R}^d$; for $l = 1, \ldots, L$ the key, query, value matrices are given by

$$\widetilde{\mathbf{K}}_{(k-1)H+h}^{(l)} = \begin{pmatrix} 0 & & & & \\ & \ddots & & & \\ & & \mathbf{K}_{k;h}^{(l)} & & \\ & & & \ddots & \\ & & & & A_k \end{pmatrix},\ \widetilde{\mathbf{Q}}_{(k-1)H+h}^{(l)} = \begin{pmatrix} 0 & & & & \\ & \ddots & & & \\ & & \mathbf{Q}_{k;h}^{(l)} & & \\ & & & \ddots & \\ & & & & I \end{pmatrix},$$

$$\widetilde{\mathbf{V}}_{(k-1)H+h}^{(l)} = \begin{pmatrix} 0 & & & & \\ & \ddots & & & \\ & & \mathbf{V}_{k;h}^{(l)} & & \\ & & & \ddots & \\ & & & & 0 \end{pmatrix},$$

$$\widetilde{\mathbf{K}}_{KH+1}^{(l)} = \begin{pmatrix} 0 & & & \\ & \ddots & & \\ & & 0 & \\ & & & A \end{pmatrix},\ \widetilde{\mathbf{Q}}_{KH+1}^{(l)} = \begin{pmatrix} 0 & & & \\ & \ddots & & \\ & & 0 & \\ & & & I \end{pmatrix},\ \widetilde{\mathbf{V}}_{KH+1}^{(l)} = \begin{pmatrix} 0 & & & \\ & \ddots & & \\ & & 0 & \\ & & & I \end{pmatrix},$$

where the submatrices $\mathbf{K}_{k;h}^{(l)}, \mathbf{Q}_{k;h}^{(l)}, \mathbf{V}_{k;h}^{(l)}$ are located in the k-th diagonal block. The output feature is given by $\widetilde{\vartheta}(y) = \begin{pmatrix} \vartheta_1(y) \\ \vdots \\ \vartheta_K(y) \\ 0 \end{pmatrix}$. Since ξ_1, ξ_m only depends on whether v_i's belong to the set Ω, the generalized position encoding pe is well-defined. We can easily verify that ϕ is indeed a general-purpose Transformer of type $(O(K), O(\log N_{\max}))$.

Let $\widetilde{X}_1^{(l)}, \ldots, \widetilde{X}_n^{(l)}$ represent the l-th hidden layer. Our goal is to show that for any $l = 1, \ldots, L$, $\widetilde{X}_i^{(l)}$ can be written as:

$$\widetilde{X}_i^{(l)} = \begin{pmatrix} X_{1;i}^{(l)} \\ \vdots \\ X_{K;i}^{(l)} \\ \widetilde{\alpha}_i \end{pmatrix},\ i = 1, \ldots, n, \tag{23}$$

where $\widetilde{\alpha}_i = \alpha_{\iota(i)} + \mathbb{1}(\iota(i) = i) \cdot \beta_{\mathcal{E}(v_i)}$ and $X_{k;i}^{(l)} \in \mathbb{R}^d$ such that

$$\|X_{k;i}^{(l)}\|_2 \leq B_\theta (KHB_v)^l. \tag{24}$$

In particular, for $i = 1, \dots, m$ we have

$$X_{k;\xi_i}^{(l)} = 0,\ \forall k = 1, \dots, K, \tag{25}$$

and for $j = 1, \dots, \xi_1$ we have

$$X_{k;j}^{(l)} = Y_{k;j}^{(l)},\ \forall k = 1, \dots, K, \tag{26}$$

and for $j = 1, \dots, \xi_1 - 1, \xi_m + 1, \dots, n$ we have

$$X_{\kappa;j}^{(l)} = Y_{\kappa, j-\xi_m-1+\xi_1}^{(l)},\ X_{k';j}^{(l)} = 0,\ \forall k' \neq \kappa, \tag{27}$$

where $Y_{k;j}^{(l)}$ is the l-th hidden layer of f_k (attending only to positions $1, \dots, \xi_1 - 1, \xi_m + 1, \dots, n)$.

Thus we apply induction on l. The case $l = 1$ holds trivially from the definition of $\widetilde{\theta}$ and $\widetilde{\mathrm{pe}}$. Suppose the above relationship holds for all layers $1, \dots, l$, consider layer $l + 1$. We have

$$\begin{aligned}
\widetilde{X}_i^{(l+1)} = & \underbrace{\sum_{k=1}^{K}\sum_{h=1}^{H}\sum_{j=1}^{i} \frac{\exp\left((\widetilde{\mathbf{Q}}_{(k-1)H+h}^{(l)}\widetilde{X}_i^{(l)})^\top(\widetilde{\mathbf{K}}_{(k-1)H+h}^{(l)}\widetilde{X}_j^{(l)})\right)}{\widetilde{Z}_{(k-1)H+h}^{(l)}} \cdot \widetilde{\mathbf{V}}_{(k-1)H+h}^{(l)}\widetilde{X}_j^{(l)}}_{\text{term 1}} \\
& + \underbrace{\sum_{j=1}^{i} \frac{\exp\left((\widetilde{\mathbf{Q}}_{KH+1}^{(l)}\widetilde{X}_i^{(l)})^\top(\widetilde{\mathbf{K}}_{KH+1}^{(l)}\widetilde{X}_j^{(l)})\right)}{\widetilde{Z}_{KH+1}^{(l)}} \cdot \widetilde{\mathbf{V}}_{KH+1}^{(l)}\widetilde{X}_j^{(l)}}_{\text{term 2}},
\end{aligned}$$

where

$$\widetilde{Z}_{(k-1)H+h}^{(l)} = \sum_{j=1}^{i} \exp\left((\widetilde{\mathbf{Q}}_{(k-1)H+h}^{(l)}\widetilde{X}_i^{(l)})^\top(\widetilde{\mathbf{K}}_{(k-1)H+h}^{(l)}\widetilde{X}_j^{(l)})\right).$$

By induction hypothesis,

$$\widetilde{X}_i^{(l)} = \begin{pmatrix} X_{1;i}^{(l)} \\ \vdots \\ X_{K;i}^{(l)} \\ \widetilde{\alpha}_i \end{pmatrix},$$

and $X_{k;i}^{(l)} = Y_{\zeta(i)}^{(l)}$ for $i = 1, \dots, \xi_1 - 1, \xi_m + 1, \dots, n$, where $\zeta(i) := \begin{cases} i, & i < \xi_1 \\ i - \xi_m - 1 + \xi_1, & i > \xi_m \end{cases}$.

Notice that for $j \leq i$:

$$\begin{aligned}
(\widetilde{\mathbf{Q}}_{(k-1)H+h}^{(l)}\widetilde{X}_i^{(l)})^\top(\widetilde{\mathbf{K}}_{(k-1)H+h}^{(l)}\widetilde{X}_j^{(l)}) &= (X_{k;i}^{(l)})^\top(\mathbf{Q}_{k;h}^{(l)})^\top\mathbf{K}_{k;h}^{(l)}X_{k;j}^{(l)} + \widetilde{\alpha}_i^\top A_k \widetilde{\alpha}_j, \\
(\widetilde{\mathbf{Q}}_{KH+1}^{(l)}\widetilde{X}_i^{(l)})^\top(\widetilde{\mathbf{K}}_{KH+1}^{(l)}\widetilde{X}_j^{(l)}) &= \widetilde{\alpha}_i^\top A \widetilde{\alpha}_j.
\end{aligned}$$

Prove Eq (23)**.** By properties of α, β, A, for any $j_2 < \xi_u < j_1 < i < \xi_{u+1}$ notice that:

$$\begin{aligned}
(\widetilde{\mathbf{Q}}_{KH+1}^{(l)}\widetilde{X}_i^{(l)})^\top(\widetilde{\mathbf{K}}_{KH+1}^{(l)}\widetilde{X}_{j_1}^{(l)}) &\geq (\widetilde{\mathbf{Q}}_{KH+1}^{(l)}\widetilde{X}_i^{(l)})^\top(\widetilde{\mathbf{K}}_{KH+1}^{(l)}\widetilde{X}_{\xi_u}^{(l)}) + C \\
&\geq (\widetilde{\mathbf{Q}}_{KH+1}^{(l)}\widetilde{X}_i^{(l)})^\top(\widetilde{\mathbf{K}}_{KH+1}^{(l)}\widetilde{X}_{j_2}^{(l)}) + 2C.
\end{aligned}$$

Due to ϵ-precision of transformers, this implies that

$$\frac{\exp\left((\widetilde{\mathbf{Q}}_{KH+1}^{(l)}\widetilde{X}_i^{(l)})^\top(\widetilde{\mathbf{K}}_{KH+1}^{(l)}\widetilde{X}_j^{(l)})\right)}{Z_{KH+1}^{(l)}} = \begin{cases} \frac{\mathbb{1}(j > \xi_u)}{i - \xi_u}, & \xi_u < i < \xi_{u+1} \\ \delta_{\xi_l}^{j}, & i = \xi_u \end{cases},$$

and hence for $\xi_u < i < \xi_{u+1}$

$$\widetilde{X}_i^{(l+1)} = \sum_{k=1}^{K}\sum_{h=1}^{H}\sum_{j=1}^{i} \frac{\exp\left((\widetilde{\mathbf{Q}}^{(l)}_{(k-1)H+h}\widetilde{X}_i^{(l)})^\top(\widetilde{\mathbf{K}}^{(l)}_{(k-1)H+h}\widetilde{X}_j^{(l)})\right)}{\widetilde{Z}^{(l)}_{(k-1)H+h}} \cdot \widetilde{\mathbf{V}}^{(l)}_{(k-1)H+h}\begin{pmatrix}\vdots\\ X_{k;j}^{(l)}\\ \vdots\\ 0\end{pmatrix}$$

$$+ \sum_{j=\xi_u+1}^{i} \cdot \frac{1}{i-\xi_u} \cdot \begin{pmatrix}0\\ \vdots\\ 0\\ \alpha_{\iota(i)}\end{pmatrix}$$

$$= \begin{pmatrix}X_{1;i}^{(l+1)}\\ \vdots\\ X_{K;i}^{(l+1)}\\ \widetilde{\alpha}_i\end{pmatrix},$$

and for $i = \xi_u$

$$\widetilde{X}_i^{(l+1)} = \sum_{k=1}^{K}\sum_{h=1}^{H}\sum_{j=1}^{i} \frac{\exp\left((\widetilde{\mathbf{Q}}^{(l)}_{(k-1)H+h}\widetilde{X}_i^{(l)})^\top(\widetilde{\mathbf{K}}^{(l)}_{(k-1)H+h}\widetilde{X}_j^{(l)})\right)}{\widetilde{Z}^{(l)}_{(k-1)H+h}} \cdot \widetilde{\mathbf{V}}^{(l)}_{(k-1)H+h}\begin{pmatrix}\vdots\\ X_{k;j}^{(l)}\\ \vdots\\ 0\end{pmatrix} + \begin{pmatrix}0\\ \vdots\\ 0\\ \alpha_{\iota(i)} + \beta_{\mathcal{E}(v_i)}\end{pmatrix}$$

$$= \begin{pmatrix}X_{1;i}^{(l+1)}\\ \vdots\\ X_{K;i}^{(l+1)}\\ \widetilde{\alpha}_i\end{pmatrix},$$

where

$$X_{k;i}^{(l+1)} = \sum_{k=1}^{K}\sum_{h=1}^{H}\sum_{j=1}^{i} \frac{\exp\left((\widetilde{\mathbf{Q}}^{(l)}_{(k-1)H+h}\widetilde{X}_i^{(l)})^\top(\widetilde{\mathbf{K}}^{(l)}_{(k-1)H+h}\widetilde{X}_j^{(l)})\right)}{\widetilde{Z}^{(l)}_{(k-1)H+h}} \cdot \mathbf{V}^{(l)}_{k;h} X_{k;j}^{(l)}. \tag{28}$$

This confirms Eq. (23) for $l+1$.

Prove Eq. (24). From above,

$$\|X_{k;i}^{(l+1)}\|_2 = \left\|\sum_{k=1}^{K}\sum_{h=1}^{H}\sum_{j=1}^{i} \frac{\exp\left((\widetilde{\mathbf{Q}}^{(l)}_{(k-1)H+h}\widetilde{X}_i^{(l)})^\top(\widetilde{\mathbf{K}}^{(l)}_{(k-1)H+h}\widetilde{X}_j^{(l)})\right)}{\widetilde{Z}^{(l)}_{(k-1)H+h}} \cdot \mathbf{V}^{(l)}_{k;h} X_{k;j}^{(l)}\right\|_2$$
$$\le KHB_v \cdot \max_{j\le i} \|X_{k;j}^{(l)}\|_2$$
$$\le B_\theta (KHB_v)^{l+1}.$$

This confirms Eq. (24) for $l+1$.

Prove Eq. (25). We first show $X_{k;\xi_1}^{(l)} = 0$. Indeed, by the properties of α_t, β_k, for any $j \le \xi_1$

$$(\widetilde{\mathbf{Q}}^{(l)}_{(k-1)H+h}\widetilde{X}_{\xi_1}^{(l)})^\top(\widetilde{\mathbf{K}}^{(l)}_{(k-1)H+h}\widetilde{X}_{\xi_1}^{(l)})$$
$$= (X_{k;\xi_1}^{(l)})^\top(\mathbf{Q}^{(l)}_{k;h})^\top\mathbf{K}^{(l)}_{k;h}X_{k;\xi_1}^{(l)} + (\alpha_0 + \beta_{\mathcal{E}(v_{\xi_1})})^\top A_k(\alpha_0 + \beta_{\mathcal{E}(v_{\xi_1})})$$
$$\ge (X_{k;\xi_1}^{(l)})^\top(\mathbf{Q}^{(l)}_{k;h})^\top\mathbf{K}^{(l)}_{k;h}X_{k;\xi_1}^{(l)} + (\alpha_0 + \beta_{\mathcal{E}(v_{\xi_1})})^\top A_k\alpha_0 + C$$
$$= (\widetilde{\mathbf{Q}}^{(l)}_{(k-1)H+h}\widetilde{X}_{\xi_1}^{(l)})^\top(\widetilde{\mathbf{K}}^{(l)}_{(k-1)H+h}\widetilde{X}_j^{(l)}) + C$$

It follows from Eq. (28) that

$$X_{k;\xi_1}^{(l+1)} = \sum_{k=1}^{K}\sum_{h=1}^{H} \mathbf{V}_{k;h}^{(l)} X_{k;\xi_1}^{(l)} = 0.$$

For ξ_i $(i > 1)$, we apply the same argument again to obtain that for any $j \le \xi_i$ such that $j \notin \{\xi_1 < \cdots < \xi_{\iota(n)}\}$ and any $i' < i$,

$$\begin{aligned}&(\widetilde{\mathbf{Q}}_{(k-1)H+h}^{(l)}\widetilde{X}_{\xi_i}^{(l)})^\top(\widetilde{\mathbf{K}}_{(k-1)H+h}^{(l)}\widetilde{X}_{\xi_{k'}}^{(l)})\\ \ge\ &(\widetilde{\mathbf{Q}}_{(k-1)H+h}^{(l)}\widetilde{X}_{\xi_1}^{(l)})^\top(\widetilde{\mathbf{K}}_{(k-1)H+h}^{(l)}\widetilde{X}_{j}^{(l)}) + C\end{aligned}$$

This implies that the attention weights are supported on $\{\xi_1 < \cdots < \xi_i\}$, and therefore

$$X_{k;\xi_i}^{(l+1)} = \sum_{k=1}^{K}\sum_{h=1}^{H}\sum_{j=1}^{i} \frac{\exp\left((\widetilde{\mathbf{Q}}_{(k-1)H+h}^{(l)}\widetilde{X}_{\xi_i}^{(l)})^\top(\widetilde{\mathbf{K}}_{(k-1)H+h}^{(l)}\widetilde{X}_{\xi_j}^{(l)})\right)}{\widetilde{Z}_{(k-1)H+h}^{(l)}} \cdot \mathbf{V}_{k;h}^{(l)} X_{k;\xi_j}^{(l)} = 0$$

where we apply the induction hypothesis $k; X_{\xi_j}^{(l)} = 0$ for all $j = 1, \ldots, i-1$. This thus completes the proof of Eq. (25).

Prove Eq. (26)**.** When $j_1 < j_2 \le i < \xi_1$, we have

$$\begin{aligned}&(\widetilde{\mathbf{Q}}_{(k-1)H+h}^{(l)}\widetilde{X}_{i}^{(l)})^\top(\widetilde{\mathbf{K}}_{(k-1)H+h}^{(l)}\widetilde{X}_{j_1}^{(l)}) - (\widetilde{\mathbf{Q}}_{(k-1)H+h}^{(l)}\widetilde{X}_{i}^{(l)})^\top(\widetilde{\mathbf{K}}_{(k-1)H+h}^{(l)}X_{j_2}^{(l)})\\ =\ &(X_{k;i}^{(l)})^\top(\mathbf{Q}_{k;h}^{(l)})^\top\mathbf{K}_{k;h}^{(l)}X_{k;j_1}^{(l)} + \alpha_0^\top A_k\alpha_0^\top\\ &-(X_{k;i}^{(l)})^\top(\mathbf{Q}_{k;h}^{(l)})^\top\mathbf{K}_{k;h}^{(l)}X_{k;j_2}^{(l)} - \alpha_0^\top A_k\alpha_0^\top\\ =\ &(\mathbf{Q}_{k;h}^{(l)}Y_{k;i}^{(l)})^\top(\mathbf{K}_{k;h}^{(l)}Y_{k;j_i}^{(l)}) - (\mathbf{Q}_{k;h}^{(l)}Y_{k;i}^{(l)})^\top(\mathbf{K}_{k;h}^{(l)}Y_{k;j_2}^{(l)}).\end{aligned}$$

It follows that

$$\widetilde{Z}_{(k-1)H+h}^{(l)} = \sum_{j=1}^{i}\exp\left((\mathbf{Q}_{k;h}^{(l)}Y_{k;i}^{(l)})^\top(\mathbf{K}_{k;h}^{(l)}Y_{k;j}^{(l)})\right),$$

and

$$\begin{aligned}X_{k;i}^{(l+1)} &= \sum_{k=1}^{K}\sum_{h=1}^{H}\sum_{j=1}^{i}\frac{\exp\left((\mathbf{Q}_{k;h}^{(l)}Y_{k;i}^{(l)})^\top(\mathbf{K}_{k;h}^{(l)}Y_{k;j}^{(l)})\right)}{\widetilde{Z}_{(k-1)H+h}^{(l)}}\cdot\mathbf{V}_{k;h}^{(l)}Y_{k;j}^{(l)}\\ &= Y_{k;i}^{(l+1)}.\end{aligned}$$

This confirms Eq. (26).

Prove Eq. (27)**.** When $i > \xi_m$, we rely on the following properties:

1. Attention sink to v_{ξ_m} for mismatch expert: for any $k' \ne \kappa$ and $j \le i$ we have
$$(\widetilde{\mathbf{Q}}_{(k'-1)H+h}^{(l)}\widetilde{X}_{i}^{(l)})^\top(\widetilde{\mathbf{K}}_{(k'-1)H+h}^{(l)}\widetilde{X}_{j}^{(l)}) \le (\widetilde{\mathbf{Q}}_{(k'-1)H+h}^{(l)}\widetilde{X}_{i}^{(l)})^\top(\widetilde{\mathbf{K}}_{(k'-1)H+h}^{(l)}\widetilde{X}_{\xi_m}^{(l)}) - C. \tag{29}$$

2. Attention to task-relevant tokens for matching expert: for $j \in \{1, \ldots, \xi_1 - 1, \xi_m + 1, \ldots, n\}$, and $\xi_1 \le j' \le \xi_m$ we have
$$(\widetilde{\mathbf{Q}}_{(\kappa-1)H+h}^{(l)}\widetilde{X}_{i}^{(l)})^\top(\widetilde{\mathbf{K}}_{(\kappa-1)H+h}^{(l)}\widetilde{X}_{j}^{(l)}) \ge (\widetilde{\mathbf{Q}}_{(\kappa-1)H+h}^{(l)}\widetilde{X}_{i}^{(l)})^\top(\widetilde{\mathbf{K}}_{(\kappa-1)H+h}^{(l)}\widetilde{X}_{j'}^{(l)}) + C. \tag{30}$$
and for $j_1 < j_2 \in \{1, \ldots, \xi - 1 - 1, \xi_m + 1, \ldots, n\}$
$$\begin{aligned}&(\widetilde{\mathbf{Q}}_{(\kappa-1)H+h}^{(l)}\widetilde{X}_{i}^{(l)})^\top(\widetilde{\mathbf{K}}_{(\kappa-1)H+h}^{(l)}\widetilde{X}_{j_1}^{(l)}) - (\widetilde{\mathbf{Q}}_{(\kappa-1)H+h}^{(l)}\widetilde{X}_{i}^{(l)})^\top(\widetilde{\mathbf{K}}_{(\kappa-1)H+h}^{(l)}\widetilde{X}_{j_2}^{(l)})\\ =\ &(\mathbf{Q}_{\kappa;h}^{(l)}Y_{\kappa;i-\xi_m-1+\xi_1}^{(l)})^\top(\mathbf{K}_{\kappa;h}^{(l)}Y_{\zeta(j_1)}^{(l)}) - (\mathbf{Q}_{\kappa;h}^{(l)}Y_{i-\xi_m-1+\xi_1}^{(l)})^\top\mathbf{K}_{\kappa;h}^{(l)}Y_{\kappa;\zeta(j_2)}^{(l)}),\end{aligned} \tag{31}$$

To see Eq. (29), we notice that

$$
\begin{aligned}
&(\widetilde{\mathbf{Q}}^{(l)}_{(k'-1)H+h}\widetilde{X}^{(l)}_i)^\top(\widetilde{\mathbf{K}}^{(l)}_{(k'-1)H+h}\widetilde{X}^{(l)}_j)\\
&=(X^{(l)}_{k';i})^\top(\mathbf{Q}^{(l)}_{k';h})^\top\mathbf{K}^{(l)}_{k';h}X^{(l)}_{k',j}+\alpha_m^\top A_{k'}(\alpha_{\iota(j)}+\beta_{\mathcal{E}(v_j)}\cdot\mathbb{1}(\iota(j)=j))\\
&\le(X^{(l)}_{k';i})^\top(\mathbf{Q}^{(l)}_{k';h})^\top\mathbf{K}^{(l)}_{k';h}X^{(l)}_{k';\xi_m}+\alpha_m^\top A_{k'}(\alpha_m+\beta_{\mathcal{E}(v_{\xi_m})})-C\\
&=(\widetilde{\mathbf{Q}}^{(l)}_{(k'-1)H+h}\widetilde{X}^{(l)}_i)^\top(\widetilde{\mathbf{K}}^{(l)}_{(k'-1)H+h}\widetilde{X}^{(l)}_{\xi_m})-C,
\end{aligned}
$$

where we use Eq. (19) with $k'\neq\kappa$.

To see Eq. (30), we notice that

$$
\begin{aligned}
(\widetilde{\mathbf{Q}}^{(l)}_{(\kappa-1)H+h}\widetilde{X}^{(l)}_i)^\top(\widetilde{\mathbf{K}}^{(l)}_{(\kappa-1)H+h}\widetilde{X}^{(l)}_j)&=(\mathbf{Q}^{(l)}_{k;h}X^{(l)}_{k;i})^\top(\mathbf{K}^{(l)}_{k;h}X^{(l)}_{k;j})+\alpha_m^\top A_\kappa\alpha_0\\
&\ge(\mathbf{Q}^{(l)}_{k;h}X^{(l)}_{k;i})^\top(\mathbf{K}^{(l)}_{k;h}X^{(l)}_{k;j'})+\alpha_m^\top A_\kappa(\alpha_{\iota(j')}+\beta_{\mathcal{E}(v_{j'})})+C\\
&=(\widetilde{\mathbf{Q}}^{(l)}_{(k-1)H+h}\widetilde{X}^{(l)}_i)^\top(\widetilde{\mathbf{K}}^{(l)}_{(k-1)H+h}\widetilde{X}^{(l)}_{j'})+C,
\end{aligned}
$$

and

$$
\begin{aligned}
(\widetilde{\mathbf{Q}}^{(l)}_{(\kappa-1)H+h}\widetilde{X}^{(l)}_i)^\top(\widetilde{\mathbf{K}}^{(l)}_{(\kappa-1)H+h}\widetilde{X}^{(l)}_j)&=(\mathbf{Q}^{(l)}_{k;h}X^{(l)}_{k;i})^\top(\mathbf{K}^{(l)}_{k;h}X^{(l)}_{k;j})+\alpha_m^\top A_\kappa\alpha_0\\
&\ge(\mathbf{Q}^{(l)}_{k;h}X^{(l)}_{k;i})^\top(\mathbf{K}^{(l)}_{k;h}X^{(l)}_{k;j'})+\alpha_m^\top A_k\alpha_{\iota(j')}+C\\
&=(\widetilde{\mathbf{Q}}^{(l)}_{(k-1)H+h}\widetilde{X}^{(l)}_i)^\top(\widetilde{\mathbf{K}}^{(l)}_{(k-1)H+h}\widetilde{X}^{(l)}_{j'})+C,
\end{aligned}
$$

where we use Eq. (20) and Eq. (22).

When $\xi_m<j_1<j_2$, Eq. (31) follows directly from

$$
\begin{aligned}
&(\widetilde{\mathbf{Q}}^{(l)}_{(\kappa-1)H+h}\widetilde{X}^{(l)}_i)^\top(\widetilde{\mathbf{K}}^{(l)}_{(\kappa-1)H+h}\widetilde{X}^{(l)}_{j_1})-(\widetilde{\mathbf{Q}}^{(l)}_{(\kappa-1)H+h}\widetilde{X}^{(l)}_i)^\top(\widetilde{\mathbf{K}}^{(l)}_{(\kappa-1)H+h}\widetilde{X}^{(l)}_{j_2})\\
&=(\mathbf{Q}^{(l)}_{k;h}X^{(l)}_{k;i})^\top(\mathbf{K}^{(l)}_{k;h}X^{(l)}_{k;j_1})+\alpha_m^\top A_k\alpha_m^\top\\
&\quad-(\mathbf{Q}^{(l)}_{k;h}X^{(l)}_{k;i})^\top(\mathbf{K}^{(l)}_{k;h}X^{(l)}_{k;j_2})+\alpha_m^\top A_k\alpha_m^\top\\
&=(\mathbf{Q}^{(l)}_{\kappa;h}Y^{(l)}_{\kappa;i-\xi_m-1+\xi_1})^\top(\mathbf{K}^{(l)}_{\kappa;h}Y^{(l)}_{j_1-\xi_m-1+\xi_1})-(\mathbf{Q}^{(l)}_{\kappa;h}Y^{(l)}_{i-\xi_m-1+\xi_1})^\top\mathbf{K}^{(l)}_{\kappa;h}Y^{(l)}_{\kappa;j_2-\xi_m-1+\xi_1}).
\end{aligned}
$$

The other cases follow similarly due to Eq. (22).

We have hence confirmed Eq. (29), Eq. (30), Eq. (31), and therefore

$$
\frac{\exp\left((\widetilde{\mathbf{Q}}^{(l)}_{(k-1)H+h}\widetilde{X}^{(l)}_i)^\top(\widetilde{\mathbf{K}}^{(l)}_{(k-1)H+h}\widetilde{X}^{(l)}_j)\right)}{\widetilde{Z}^{(l)}_{(k-1)H+h}}=\begin{cases}\delta_j^{\xi_m}, & k\neq\kappa\\ \frac{\exp\left((\mathbf{Q}^{(l)}_{\kappa;h}Y^{(l)}_{\kappa;i-\xi_m-1+\xi_1})^\top(\mathbf{K}^{(l)}_{\kappa;h}Y^{(l)}_j)\right)}{\widetilde{Z}^{(l)}_{(k-1)H+h}}, & k=\kappa,\ j<\xi_1\\ 0, & k=\kappa,\ \xi_1\le j\le\xi_m\\ \frac{\exp\left((\mathbf{Q}^{(l)}_{\kappa;h}Y^{(l)}_{\kappa;i-\xi_m-1+\xi_1})^\top(\mathbf{K}^{(l)}_{\kappa;h}Y^{(l)}_{j-\xi_m-1+\xi_1})\right)}{\widetilde{Z}^{(l)}_{(k-1)H+h}}, & k=\kappa,\ j>\xi_m\end{cases}
$$

and

$$
\widetilde{Z}^{(l)}_{(k-1)H+h}=\sum_{j=1,\ldots,\xi_1-1,\xi_m+1,\ldots,n}\exp\left((\mathbf{Q}^{(l)}_{\kappa;h}Y^{(l)}_{\kappa;i-\xi_m-1+\xi_1})^\top(\mathbf{K}^{(l)}_{\kappa;h}Y^{(l)}_j)\right).
$$

It follows that

$$
\begin{aligned}
X^{(l+1)}_{\kappa;i}&=\sum_{j=1}^{\xi_1-1}\frac{\exp\left((\mathbf{Q}^{(l)}_{\kappa;h}Y^{(l)}_{\kappa;i-\xi_m-1+\xi_1})^\top(\mathbf{K}^{(l)}_{\kappa;h}Y^{(l)}_j)\right)}{\widetilde{Z}^{(l)}_{(k-1)H+h}}\mathbf{V}^{(l)}_{k;h}Y^{(l)}_j\\
&\quad+\sum_{j=\xi_m+1}^{i}\frac{\exp\left((\mathbf{Q}^{(l)}_{\kappa;h}Y^{(l)}_{\kappa;i-\xi_m-1+\xi_1})^\top(\mathbf{K}^{(l)}_{\kappa;h}Y^{(l)}_{j-\xi_m-1+\xi_1})\right)}{\widetilde{Z}^{(l)}_{(k-1)H+h}}\mathbf{V}^{(l)}_{k;h}Y^{(l)}_{j-\xi_m-1+\xi_1},\\
&=Y^{(l+1)}_{\kappa;i-\xi_m-1+\xi_1}\\
X^{(l+1)}_{k';i}&=X^{(l)}_{k';\xi_m}=0,\ \forall k'\neq\kappa.
\end{aligned}
$$

Therefore we establish Eq. (27). This completes the induction.

At the output layer, we have

$$\begin{aligned}p_{\widetilde{f}}(y|v_1,\dots,v_n) &= \mathrm{Softmax}(\widetilde{\vartheta}(y)^\top \widetilde{X}_n^{(L)})\\ &= \mathrm{Softmax}(\vartheta(y)^\top Y_{n-\xi_m-1+\xi_1}^{(L)})\\ &= p_{f_\kappa}(y|u_1,\dots,u_{n-\xi_m-1+\xi_1}).\end{aligned}$$

This establishes the desired Eq. (2). □

A.5 Proof of Theorem 4.7

Proof. Let ϕ_s, ϕ_m, ϕ_e denote the general-purpose Transformers in Proposition 4.4 (with K experts), 4.2 (with $K = 3$ token spaces), and A.1 (extending to $\mathcal{V}$) respectively. We construct a dummy Transformer f_d that outputs BOS immediately after a token in $\mathcal{A}$. Then we claim that the general-purpose Transformer $\widetilde{\phi}$ defined by

$$\widetilde{\phi}(f_0, f_1, \dots, f_K) = \phi_m(\phi_s(\phi_e(f_1), \dots, \phi_e(f_K)), f_d, f_0)$$

achieves the desired property.

Indeed, let $g_1 = \phi_s(\phi_e(f_1), \dots, \phi_e(f_K))$, by Proposition 4.4, we have

1. **Expert following**: At t-th iteration,
$$p_{g_1}\left(\cdot\middle|\text{prompt}\right) \sim p_{f_{a^{(t)}}}\left(\cdot\middle|q|u_{1:i-1}^{(t)}\right),$$
where $q|u_{1:i-1}^{(t)}$ is the token sequence obtained by concatenating the user query q and prior generated part in response t: $u_{1:i-1}^{(t)}$.

2. **Regret minimization**:
$$\max_{a^*\in\mathcal{A}} r_0(a^*) - \mathbb{E}[r_0(a^{(T)})] \le \mathrm{reg}(T).$$

Therefore by Proposition 4.2, we have

$$u_i^{(t)} \sim p_{f_{a^{(t)}}}\left(\cdot\middle|q|u_{1:i-1}^{(t)}\right).$$

It follows that

$$\begin{aligned}\max_{u^*\in\mathcal{V}^\omega} r(q,u^*) - \mathbb{E}[r(q,u^{(T)})] &\le \lambda + \mathbb{E}_{u\sim f_{k^*}(\cdot|p)}[r(q,u)] - \mathbb{E}_{a^{(T)}}\left[\mathbb{E}_{u^{(T)}\sim f_{a^{(t)}}(\cdot|q)}[r(q,u^{(T)})]\right]\\ &\le \lambda + \max_{a^*\in\mathcal{A}} r_0(a^*) - \mathbb{E}[r_0(a^{(T)})]\\ &\le \lambda + \mathrm{reg}(T).\end{aligned}$$

Finally, $\widetilde{\phi}$ has type ϕ of type $(O(K), O(\log(N_{\max})))$ because ϕ_s has type $(O(K), O(\log(N_{\max})))$ and ϕ_m, ϕ_e has type $(O(1), O(\log(N_{\max})))$. This completes the proof. □

A.6 Attention Sink Positional Encoding

In this section, we introduce positional encoding mechanisms that induce attention sink behaviors used by Theorem 4.7.

Lemma A.2 (Attention Sink Positional Encoding, Type 1)**.** *For any $C \in \mathbb{R}_+$, $K, N \in \mathbb{Z}_+$, there exist vectors $\alpha_1, \dots, \alpha_N, \beta_1, \dots, \beta_K \in \mathbb{R}^d$ and matrices $A, A_1, \dots, A_K \in \mathbb{R}^{d\times d}$ for $d \le O(K + \log N)$ such that for any $n \in [N]$ the followings hold*

1. *For any $k \ne k'$:*
$$\alpha_n^\top A_k(\alpha_n + \beta_{k'}) \ge C + \begin{cases}\alpha_n^\top A_k \alpha_n\\ \alpha_n^\top A_k\alpha_j\\ \alpha_n^\top A_k(\alpha_j + \beta_{k''})\end{cases}, \forall 0 \le j \le n, 1 \le k'' \le K.$$

2. *For any* $k \in [K]$*:*

$$\alpha_n^\top A_k \alpha_n = \alpha_n^\top A_k \alpha_0 \geq C + \begin{cases} \alpha_n^\top A_k(\alpha_n + \beta_k) \\ \alpha_n^\top A_k \alpha_j \\ \alpha_n^\top A_k(\alpha_j + \beta_{k'}) \end{cases}, \ \forall 0 < j < n, k' \neq k.$$

3. *For any* $k, k', k'' \in [K]$*:*

$$(\alpha_n + \beta_{k'})^\top A_k(\alpha_n + \beta_{k'}) \geq C + (\alpha_n + \beta_{k'})^\top A_k \alpha_j, \ \forall 0 \leq j \leq n.$$

4. *For any* $0 < j < n$*:*

$$\begin{aligned} \alpha_n^\top A \alpha_n &\geq \alpha_n^\top A(\alpha_n + \beta_k) + C \\ &\geq C + \max\{\alpha_n^\top A \alpha_j, \alpha_n^\top A(\alpha_j + \beta_{k'})\}, \ \forall k, k'' \in [K]. \end{aligned}$$

Proof. Notice that the following relations are sufficient to guarantee the desired properties

$$\begin{aligned} \alpha_n^\top A_k \alpha_n &= \alpha_n^\top A_k \alpha_0, \\ \alpha_n^\top A_k \beta_{k'} &= C, \\ \alpha_n^\top A_k \alpha_n &\geq \alpha_n^\top A_k \alpha_j + \alpha_n^\top A_k \beta_{k'} + C, \\ \alpha_n^\top A_k \beta_k &= -C, \\ \alpha_n^\top A \beta_k &= -C, \\ \beta_{k'}^\top A_k \beta_{k'} &= 9C. \end{aligned}$$

By Lemma A.4, we can find $\gamma_1, \ldots, \gamma_N \in \mathbb{R}^{\bar{d}}$ such that $\bar{d} = O(\log N)$, $\gamma_i^\top \gamma_j \leq 1/2$ for any $i \neq j \in [N]$, and $\gamma_i^\top \gamma_i \geq 1$ for any $i \in [N]$. Define

$$B_k = e_k e_k^\top, \ \eta_k = -e_k.$$

where $e_1, \ldots, e_K$ form the standard basis of $\mathbb{R}^K$.

We thus let

$$\alpha_i = \begin{pmatrix} a\gamma_i \\ b\mathbf{1}_E \\ c1 \\ c1 \\ 0 \end{pmatrix}, \ \beta_k = \begin{pmatrix} 0 \\ f\eta_k \\ e \\ -e \\ h \end{pmatrix}, \ \alpha_0 = \begin{pmatrix} 0 \\ 0 \\ g1 \\ -g1 \\ 0 \end{pmatrix}$$

$$A_k = \begin{pmatrix} I & & & & \\ & B_k & & & \\ & & 1 & & \\ & & & -1 & \\ & & & & 1 \end{pmatrix}, \ A = \begin{pmatrix} I & & & & \\ & I/K & & & \\ & & 0 & & \\ & & & 0 & \\ & & & & 0 \end{pmatrix},$$

where $b = c = f = \sqrt{C}, e = \sqrt{C}/2, a = \sqrt{3C}, g = 2\sqrt{C}, h = 3\sqrt{C}$. The dimension can be bounded by $d = \bar{d} + K + 3 = O(K + \log N)$. □

Lemma A.3 (Attention Sink Positional Encoding, Type 2)**.** *For any* $C \in \mathbb{R}_+$*,* $K, N \in \mathbb{Z}_+$*, there exist vectors* $\alpha_1, \ldots, \alpha_N, \beta_0, \ldots, \beta_K \in \mathbb{R}^d$ *and matrices* $A, A_1, \ldots, A_K \in \mathbb{R}^{d \times d}$ *for* $d \leq O(K + \log N)$ *such that for any* $n \in [N]$ *the followings hold*

1. *For any* $i \geq j_1, j_2, j_3$ *and* $k, k', k'' \neq 0$*:*

$$(\alpha_i + \beta_k)^\top A_0(\alpha_{j_1} + \beta_{k'}) = (\alpha_i + \beta_k)^\top A_0(\alpha_{j_2} + \beta_{k''}) \geq (\alpha_i + \beta_k)^\top A_0(\alpha_{j_1} + \beta_0) + C$$

$$(\alpha_i + \beta_0)^\top A_0(\alpha_i + \beta_0) \geq (\alpha_i + \beta_0)^\top A_0(\alpha_{j_1} + \beta_k) + C.$$

2. *For any* $i > j$ *and* $k \neq k' \neq 0$

$$\begin{aligned} (\alpha_i + \beta_k)^\top A(\alpha_i + \beta_k) &\geq (\alpha_i + \beta_k)^\top A(\alpha_j + \beta_{k'}) + C \\ &\geq (\alpha_i + \beta_k)^\top A(\alpha_j + \beta_0) + 2C. \end{aligned}$$

3. *For any $i \geq j, j_1$ and $k \neq k', k''$*

$$(\alpha_i + \beta_k)^\top A_{k'}(\alpha_j + \beta_0) \geq (\alpha_i + \beta_k)^\top A_{k'}(\alpha_{j_1} + \beta_{k''}) + C$$

$$(\alpha_i + \beta_k)^\top A_k(\alpha_i + \beta_k) \geq \max\{(\alpha_i + \beta_k)^\top A_k(\alpha_{j_1} + \beta_{k''}), (\alpha_i + \beta_k)^\top A_{k'}(\alpha_{j_1} + \beta_0)\} + C.$$

Proof. Following the notations in Lemma A.2, let

$$\alpha_i = \begin{pmatrix} \gamma_i \\ 0 \\ 0 \\ 0 \end{pmatrix}, \beta_k = \begin{pmatrix} 0 \\ \gamma \\ e_k \\ 1 \end{pmatrix}, \beta_0 = \begin{pmatrix} 0 \\ \gamma \\ 1 \\ f \end{pmatrix},$$

and

$$A = \begin{pmatrix} 0 & & & \\ & a \cdot I & & \\ & & 0 & \\ & & & 0 \end{pmatrix}, A_k = \begin{pmatrix} b \cdot I & & & \\ & 0 & & \\ & & c \cdot e_k e_k^\top & \\ & & & 1 \end{pmatrix}, A = \begin{pmatrix} e \cdot I & & & \\ & 0 & & \\ & & 0 & \\ & & & 0 \end{pmatrix},$$

where $a = c = e = C, f = 3.5C, d = 4C$. The dimension can be bounded by $d = \bar{d} + K + 3 = O(K + \log N)$. □

A.7 Technical Claims

Claim A.4 (Johnson-Lindenstrauss Lemma)**.** Given $0 < \varepsilon < 1$, a set X of N points in $\mathbb{R}^n$, and an integer $k > \frac{8(\ln N)}{\varepsilon^2}$, there is a linear map $f : \mathbb{R}^n \to \mathbb{R}^k$ such that

$$(1 - \varepsilon)\|u - v\|^2 \leq \|f(u) - f(v)\|^2 \leq (1 + \varepsilon)\|u - v\|^2$$

holds for all $u, v \in X$.

Claim A.5 (Concentration of Multinomial Distributions, adapted from Agrawal & Jia (2017))**.** Let $p \in \Delta^S$ and $\hat{p} \sim \frac{1}{n}\text{Multinomial}(n, p)$. Then, for any $\delta \in [0, 1]$:

$$\mathbb{P}\left(\|\hat{p} - p\|_1 \geq \sqrt{\frac{2\ln(1/\delta)}{n}}\right) \leq \delta.$$

Claim A.6 (Berry-Esseen theorem)**.** If $X_1, X_2, \ldots$ are i.i.d. random variables with $\mathbb{E}(X_1) = 0$, $\mathbb{E}(X_1^2) = \sigma^2 > 0$, and $\mathbb{E}(|X_1|^3) = \rho < \infty$, we define

$$Y_n = \frac{X_1 + X_2 + \cdots + X_n}{n}$$

as the sample mean, with F_n the cumulative distribution function of $\frac{Y_n\sqrt{n}}{\sigma}$ and Φ the cumulative distribution function of the standard normal distribution, then for all x and n,

$$|F_n(x) - \Phi(x)| \leq \frac{8\rho}{\sigma^3\sqrt{n}}.$$

B Experiment Details

B.1 Implementation details of self-correction experiments

The model configurations are detailed in Table 2. Our code is implemented based on `PyTorch` Paszke et al. (2019) and `minGPT`[2]. All the models are trained on one NVIDIA GeForce RTX 2080 Ti GPU with 11GB memory.

Model	**Depth**	**Heads**	**Width**
GPT-nano	3	3	48
GPT-micro	4	4	128
GPT-mini	6	6	192
Gopher-44M	8	16	512

Table 2: Model configuration hyperparameters.

Following common practice, the learning rate goes through the warm-up stage in the first 5% of training iterations, and then decays linearly to 0 until training finishes. We set the peak learning rate to 10^{-4} and find that all the models are stably trained under this learning rate schedule. We do not apply drop out or weight decay during training. We repeat the experiments for 3 times under different random seeds and report the average accuracy with error bars.

B.2 Prompts for self-correction

Initial Problem Solving Prompt

Solve the following math problem efficiently and clearly. The last line of your response should be of the following format: 'Therefore, the final answer is: `$\boxed{ANSWER}$`. I hope it is correct' (without quotes) where `ANSWER` is just the final number or expression that solves the problem. Think step by step before answering.
{Question}

Correction Prompt

Your answer is incorrect. Please analyze your solution and identify where you made an error. Then provide a corrected solution that leads to the right answer. The last line of your response should be of the following format: 'Therefore, the final answer is: `$\boxed{ANSWER}$`.'

C Limitations

Despite these contributions, our work comes with limitations: our construction in Theorem 4.7 only applies to attention-only Transformers and relies on a slightly generalized position encoding method. Relaxing these constraints constitutes interesting problems for future research.

Large Language Models Usage Disclore

LLMs were used only to polish writing.

[2] `https://github.com/karpathy/minGPT` (MIT license).