Title

Large Language Models as Statistical Decision-Makers at Inference Time

ISBN

9798189270154

Author

Huang, Baihe

Publication Date

2026-08-01

Peer reviewed|Thesis/dissertation

Large Language Models as Statistical Decision-Makers at Inference Time

by

Baihe Huang

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Electrical Engineering and Computer Sciences

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Michael I. Jordan, Chair
Assistant Professor Jiantao Jiao
Professor Ryan Tibshirani

Summer 2026

Large Language Models as Statistical Decision-Makers at Inference Time

Abstract

Large Language Models as Statistical Decision-Makers at Inference Time

by

Baihe Huang

Doctor of Philosophy in Electrical Engineering and Computer Sciences

University of California, Berkeley

Professor Michael I. Jordan, Chair

This dissertation studies Large Language Model (LLM) inference through the lens of statistical decision-making theory, with the goal of advancing prevailing inference paradigms along two principal axes: scalability and provenance.

In the domain of scalability, this dissertation investigates the efficiency of inference-time scaling paradigms and parallel decoding schemes. We formally characterize the sample complexity of self-consistency and best-of-n sampling methodologies, and demonstrate how self-correction can substantially expand the expressive power of Transformer architectures in multi-task settings. We analyze the information-theoretic bottlenecks associated with parallel sampling in diffusion language models, and introduce Explore-Then-Exploit, a scalable decoding strategy that improves inference throughput while preserving generation quality.

In the domain of provenance, the dissertation develops theoretical foundations and practical schemes for statistical watermarking. We mathematically formalize statistical watermarking as a hypothesis-testing problem, providing a comprehensive characterization of the optimal Type II error, token efficiency, and robustness against perturbations. Guided by the theoretical insights, we propose SEAL, a semantic-aware watermarking scheme demonstrating strong detection efficiency and tamper resistance. We further generalize the statistical watermarking framework to support anytime-valid detection and derive the optimal e-value-based detection scheme.

Collectively, our work connects theoretical insights with algorithmic innovations for understanding and improving the efficiency and responsible use of LLM inference.

To my parents.

Contents

List of Figures

List of Tables

Acknowledgments

First and foremost, I would like to express my deepest gratitude to my advisors, Professor Michael I. Jordan and Professor Jiantao Jiao. Their invaluable guidance and unwavering support throughout my Ph.D. have profoundly shaped both the research within this dissertation and my overall development as an independent scholar. I am deeply thankful to Mike for his insightful comments, career advice, and the rich opportunities he provided to travel and connect with remarkable researchers worldwide. I always walked away from our conversations having learned something new; Mike's broad vision and vast knowledge consistently expanded my academic horizons and illuminated the deep connections between disparate fields. I am immensely grateful to Jiantao for introducing me to the field of large language models. His sharp vision and research guidance consistently pointed us toward the cutting edge, making our discussions incredibly inspiring and fruitful. I am also tremendously appreciative of the industry opportunities Jiantao facilitated and his steadfast support during my job search.

I would like to extend my sincere gratitude to Professor Ryan Tibshirani for serving on my dissertation committee; his thoughtful commentary and constructive feedback significantly elevated the quality of this dissertation. I am deeply thankful to Professor Jason D. Lee for offering invaluable mentorship at the early stages of my academic journey and consistent support over the years. It has been an absolute pleasure discussing ideas with Jason and learning from his precise insights into machine learning theory. Additionally, I would like to thank Professors Kannan Ramchandran, Federico Echenique, and Stuart J. Russell, whose inspiration and influence on my work and thinking have been profound.

Research is rarely a solitary endeavor, and I've been incredibly fortunate to work with many brilliant collaborators. I want to thank all of them, especially my close collaborators Hanlin Zhu, Banghua Zhu, Sai Praneeth Karimireddy, Shanda Li, Tianhao Wu, and Hengyu Fu, for their wonderful work and partnership.

Beyond academia, I had the privilege of learning from outstanding mentors during my industry internships. At Microsoft Research, I am thankful to Hiteshi Sharma for her concrete advice and support for my research on large language models. At Voleon, I am grateful to Max Zimet for his thoughtful guidance and stimulating comments on my project. I am also thankful to Venkat Srinivasan at NVIDIA for the engaging discussions and fruitful collaborations on numerous exciting projects.

I want to thank my friends Tianyu Guo, Hanlin Zhu, Licong Lin, Ruiqi Zhang, Yuhang Cai, Hengyu Fu, Jiachen Lian, Yuzheng Hu, Tingle Li, and Mingxuan Cai for making this journey full of joy, laughter, and unforgettable memories. I am appreciative of Geng Zhao, Alireza Fallah, Jordan Lekeufack Sopze, Xuelin Yang, Serena Wang, Wenshuo Guo, Tianyi Lin, Eric Zhao, Anastasios N. Angelopoulos, Ian Waudby-Smith, Liviu Aolaritei, Xinyan Hu, Stephen Bates, Druv Pai, Zixuan Wang, Haozhe Jiang, Yilong Hou, Eric Xu, Kunhe Yang, Paria Rashidinejad, Nived Rajaraman, and all my labmates at SAIL, Jiao lab, and BLISS lab for cultivating such a supportive and welcoming environment. I must also thank the staff of the EECS department, especially Judy Ileana Smithson, Naomi Yamasaki, Jean Nguyen, Shirley Salanio, and Susanne Kauer, for their extensive help and patience.

Finally, and most importantly, my deepest love and gratitude go to my mom and dad. None of this would have been possible without your unconditional love.

Chapter 1

Introduction

In recent years, Large Language Models (LLMs) have achieved remarkable success across a broad spectrum of domains, encompassing mathematical reasoning [52, 53, 64, 141, 147], code generation [13, 34, 89, 113], multimodal understanding and generation [121, 145, 156, 157], personal assistance [148, 166, 201, 218], healthcare [140, 167, 177, 178], and scientific discovery [12, 25, 65, 146]. Evaluations of frontier LLM agents on software-engineering tasks, indexed by the human time required for completion, demonstrate exponential growth in the task horizons these agents can successfully navigate [104].

These breakthroughs are propelled not solely by scaling model parameters, but also by the development of increasingly sophisticated inference procedures, including test-time scaling [22, 179, 204], agentic tool-use [166, 218], and structured harnesses [28, 95, 106, 208]. Meanwhile, the rapid growth of LLM inference capability introduces a set of novel theoretical and practical challenges:

- First, classical theoretical frameworks are no longer adequate for capturing complex inference procedures. Traditionally, we treat machine learning models, including LLMs, as simple input-output functions. However, modern LLM systems execute long-horizon reasoning [45, 144], leverage tools to interact with diverse environments [108, 136, 211, 215], and operate across multiple turns [219, 237]. This paradigm shift necessitates novel theoretical instruments capable of characterizing not just the black-box model, but the intricate inference-time procedures built around it.

- Second, as LLM outputs become increasingly indistinguishable from human writing, numerous societal risks of LLM content misuse emerge. Prior literature highlights the role of LLMs in academic misconduct and educational integrity [87, 132]. Large-scale analyses of AI conference peer reviews and policies similarly underscore threats to the scientific review process [115, 163]. Furthermore, there is growing concern that AI-generated data may contaminate the training corpora, thereby degrading the scaling of future models [43, 175]. Collectively, these vulnerabilities call for principled methodologies that establish provenance and mitigate the misuse of LLM inference.

Motivated by these challenges, this dissertation is centered on the following core research question: *How can we design theoretical principles and practical methodologies for understanding and improving the efficiency and responsible use of LLM inference?* We address this question through a unifying methodology that casts inference as a statistical decision-making procedure, grounded in the next-token prediction paradigm underlying modern autoregressive language models [1, 23, 73, 92]. Through the lens of distributional natural language modeling, inference-time interventions such as test-time scaling, repeated sampling, feedback-based scoring, self-correction, and watermarking are formalized as statistical decision-making under uncertainty. Accordingly, this dissertation investigates LLM inference by formalizing the relevant inference-time procedures, characterizing their fundamental efficiency and validity, and utilizing the theoretical insights to guide algorithm design.

1.1 Contributions

This dissertation focuses on developing statistical frameworks to understand and improve the inference paradigms of LLMs along two primary dimensions: scalability and provenance.

Part I: Scalability.

- **Chapter 2: Sample complexity and expressiveness of inference-time scaling paradigms.** Recent work by Snell et al. [179] demonstrates that increasing compute during inference can substantially enhance LLM performance on complex reasoning tasks such as mathematical problem-solving, a phenomenon formally known as inference-time or test-time scaling. This scaling operates along two primary axes: increasing the number of independent repeated samples, and extending the length of the response trajectory. In this chapter, we analyze the sample complexity and expressive power of three representative methodologies that anchor these axes: *self-consistency* (which samples n i.i.d. responses and selects the most frequent answer, marginalizing over intermediate reasoning paths), *best-of-n* (which samples n i.i.d. responses but selects the one maximizing a given verifier score), and *self-correction* (where the LLM autonomously generates a sequence of n responses, each conditioned on prior iterations and their associated scores). For repeated sampling schemes, we establish a fundamental separation in sample complexity between self-consistency and best-of-n. For self-correction, we provide a representation-theoretic result demonstrating its expressiveness in multi-task problem-solving. This chapter is based on joint work with Shanda Li, Tianhao Wu, Yiming Yang, Ameet Talwalkar, Kannan Ramchandran, Michael I. Jordan, and Jiantao Jiao [81].

- **Chapter 3: Bits-to-rounds sampling for diffusion language model inference.** Diffusion language models offer a promising alternative to autoregressive language models by allowing multiple tokens to be unmasked in one round, bypassing the left-to-right generation order. However, the scalability of their inference speed remains

bottlenecked by the "curse of parallel decoding," wherein tokens sampled simultaneously during parallel decoding are (conditionally) independent and thus deviate from the true joint distribution [207]. We establish an information-theoretic lower bound for confidence-based decoding schemes that associates the required number of decoding rounds with total information and the dynamic confidence threshold serving as a per-round information budget. Guided by this theoretical bound, we introduce *Explore-Then-Exploit* (ETE), a training-free decoding algorithm that harmonizes cross-block parallel decoding with targeted, shared-KV beam-search exploration of high-information tokens. ETE reduces the number of decoding rounds by 26–61% and improves wall-clock throughput by up to 70%, while maintaining matched accuracy on diverse benchmarks including MMLU-Pro, GSM8K, MATH, and HumanEval. This chapter is based on joint work with Hengyu Fu, Virginia Adams, Charles Wang, Junkeun Yi, Mohammad Mahdi Kamani, Venkat Srinivasan, and Jiantao Jiao [56].

Part II: Provenance.

- **Chapter 4: Theoretical foundations for statistical watermarking.** Statistical watermarking [2, 99] provides a principled mechanism for establishing the provenance of LLM inference, thereby demonstrating a promising path to mitigate the risks of malicious or irresponsible use of AI-generated content. In this chapter, we formalize statistical watermarking as an active hypothesis-testing problem featuring random, output-coupled rejection regions. We derive the Uniformly Most Powerful (UMP) watermark and obtain its closed-form Type-II error. For model-agnostic watermarks, we establish the (nearly) minimax optimal excess Type-II error relative to the corresponding UMP watermark with the same Type-I error level. To understand the token efficiency required to achieve prescribed error rates, we establish nearly matching upper and lower bounds on the number of i.i.d. tokens in terms of the average entropy per token. Finally, we characterize the robustness of watermarking against bounded perturbations by formulating the optimal Type-II error as the solution to an explicit linear program. This chapter is based on joint work with Hanlin Zhu, Banghua Zhu, Kannan Ramchandran, Michael I. Jordan, Jason D. Lee, and Jiantao Jiao [79].

- **Chapter 5: Semantic-aware statistical watermarking.** Our theoretical analysis reveals that the detection power of a watermark fundamentally depends on the tightness of the coupling between the random seed sequence and the generated output sequence. This insight exposes a critical limitation in existing watermarking methodologies, which rely on fixed seed distributions that fail to adapt to the LLM's dynamic token probabilities. Addressing this gap, we introduce **SEAL** (**S**emantic-awar**E** specul**A**tive samp**L**ing), a practical watermarking algorithm that constructs a maximal coupling between the random seed sequence and output tokens via speculative sampling [107] using a publicly accessible proposal LLM. By leveraging the inherent structural similarities among LLMs trained on shared human-language corpora, SEAL generates

seeds that are semantic-aware and thus reconciles the deployability requirements of model-agnosticism with the sample efficiency predicted by our theoretical framework. Evaluated on the MarkMyWords benchmark [152], SEAL demonstrates superior token efficiency and tamper resistance, especially against paraphrasing attacks. This chapter is based on joint work with Hanlin Zhu, Julien Piet, Banghua Zhu, Jason D. Lee, Kannan Ramchandran, Michael I. Jordan, and Jiantao Jiao [80].

- **Chapter 6: Anytime-valid statistical watermarking.** While watermark generation is inherently sequential and variable in length due to the autoregressive nature of LLMs, conventional watermark detection requires ad hoc, fixed-horizon testing. In this paradigm, applying the p-value test repeatedly as tokens arrive violates the Type-I error guarantee, thereby limiting detection efficiency and precluding early stopping. We bridge this crucial operational gap by proposing *Anchored E-Watermarking*, an e-value-based statistical watermarking framework that allows anytime-valid detection and early stopping. Within this framework, we derive the closed-form minimax optimal logarithmic growth rate of evidence and establish a tight bound on the expected detection time. Empirical evaluations on the MarkMyWords benchmark [152] in the sequential testing setting confirm that the resulting detector reduces the median token budget required for detection by up to 15% relative to SEAL, and by nearly 50% compared to the other existing baselines. This chapter is based on joint work with Eric Xu, Kannan Ramchandran, Jiantao Jiao, and Michael I. Jordan [82].

1.2 Related Work

Theories of Transformers and Large Language Models. The success of Transformers and LLMs has motivated the study of their expressiveness. Existing research has shown that Transformers can implement simple functions such as sparse linear functions, two-layer neural networks, and decision trees [59], gradient descent [8, 16, 191], automata [120, 233], Dyck languages [18, 217], Turing machines [19, 46, 150, 203, 225], variational inference [130], and bandit algorithms [117]. Alberti et al. [9], Luo et al. [126], Petrov et al. [151], Yun et al. [223] establish universal approximation results under various settings. Edelman et al. [47], Li et al. [109], Likhosherstov et al. [116], Nelson Elhage et al. [137] study representational capabilities and properties of self-attention, the core component in Transformers. Feng et al. [50], Li et al. [114] study the expressiveness of auto-regressive Transformers with chain-of-thought. Botta et al. [20], Edelman et al. [47], Li et al. [112] study 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. [182] introduce the generation-verification gap as a key quantity governing scaling behavior. Huang et al. [78] propose a progressive sharpening framework in which the policy gradually shifts toward more confident responses. Setlur et al. [169] draw 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 test-time scaling laws for LLMs, illuminating a new scaling axis beyond training-time scaling laws [73, 92]. Existing approaches towards scaling up test-time compute of LLMs can be broadly classified into two categories: (1) applying test-time algorithms, also called inference-time algorithms, during LLM decoding [22, 179, 209]; and (2) explicitly training LLMs to output long chain-of-thought traces [45, 97, 143, 212]. Many recent works focus on understanding and improving the effectiveness of test-time scaling empirically: Aggarwal and Welleck [6], Chen et al. [35], Cuadron et al. [42], Wang et al. [200] study under-thinking, over-thinking, and length control in LLM reasoning. Chen et al. [32] propose to integrate self-verification and self-correction into sampling. Qu et al. [155] analyze test-time compute optimization by introducing a meta reinforcement learning formulation. Setlur et al. [169] demonstrate that verification/RL is important for optimal test-time scaling. Zhang et al. [230] 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 self-correction analysis offers a formal perspective on In-Context Reinforcement Learning [134, 180, 187].

Diffusion Language Models. Discrete diffusion models have been widely adopted for categorical data generation, including natural language [139], biological sequences [164], code synthesis [176], and audio generation [214]. The framework was first proposed by Austin et al. [15] as a discrete variant of Denoising Diffusion Probabilistic Models (DDPM) [72], and later reformulated as a probability ratio learning problem [125]. Early models typically operated at small scales (fewer than 1B parameters). Through reparameterization [238] and engineering efforts such as KV-caching [123], masked (absorbing) diffusion models have emerged as the predominant paradigm due to their scalability. This scalability has enabled the development of diffusion-based large language models (DLMs) [139, 220, 224, 239], which achieve performance and inference speed comparable to autoregressive (AR) language models [153]. Recent works have also explored hybrid approaches through block-wise diffusion [14, 29], combining the flexible content/length generation of AR models with the parallel inference capabilities of DLMs.

Parallel Decoding in DLMs. Unlike autoregressive language models that generate tokens sequentially from left to right, DLMs frame generation as an iterative denoising (unmasking) process over entire sequences, and enable parallel decoding through bidirectional attention. Early approaches adopted fixed-step parallel decoding [27, 139], while more recent methods introduce adaptive strategies. Yu et al. [222] propose confident decoding, which dynamically unmasks high-confidence tokens above a fixed threshold. Wu et al. [207] extend this with a dynamic confidence threshold mechanism, and Wei et al. [205] introduce a two-

phase approach, which alternates between two decoding stages, performing conservative decoding until high-confidence regions form and then conducting aggressive confidence-based parallel decoding on those confident regions. Huang et al. [83] analyze decoding biases through confidence calibration, while Ben-Hamu et al. [17] leverage entropy-based unmasking with controlled approximation error. Kim et al. [96] introduce the probability margin as the confidence measure. Additional acceleration techniques include attention optimization via caching mechanisms [123] and sparsification [224]. However, all these approaches share a common principle: prioritizing certain (high-confidence/low-entropy) tokens first. This strategy inherently reduces the information revealed per inference step, potentially limiting overall decoding speed from an information-theoretic perspective, given that the total information content is fixed. Notably, recent findings in Reinforcement Learning with Verifiable Rewards (RLVR) suggest that uncertain (high-entropy) tokens are key to successful and efficient reasoning [197], which contrasts with the current certain-token-first paradigm in DLM decoding. ETE is complementary to the line of work on lossless speculative decoding for DLMs, such as Spiffy [7]. Rather than using a separate draft model or an auto-speculative draft graph, ETE uses two-step anchor-based decoding as the proposal and an information-theoretic score function as the verification.

Statistical watermarking. Watermarking is a powerful white-box provenance approach for detecting LLM-generated texts [185] in which the model provider modifies the generation so that a verifier can later test for the embedded signal. Watermarks can be injected either into a pre-existing text (edit-based watermarks) or during the text generation (generative watermarks), while our work falls in the latter category. Edit-based watermarking [4, 91, 160, 216] has been the focus of several studies in the past. The concept of generative watermarking dates back to the work of Venugopal et al. [190], while our work is more related to the seminal work by Aaronson and Kirchner [3] and Kirchenbauer et al. [99] which introduced statistical watermarking as a provable method of embedding statistical signals into language model generation. To develop formal guarantees, Kuditipudi et al. [102] introduce the notion of distortion-free watermarking and inverse transform sampling as a new watermarking method. Following Kirchenbauer et al. [99], several works [57, 74, 75, 122, 159] leverage the semantics of preceding tokens to determine the green list and adjust the bias applied to green-list tokens. Specifically, Hou et al. [74, 75] use the partition of semantic space to serve as green and red lists for sentence embeddings and perform rejection sampling to sample the sentences conditional on a particular green region in the semantic space; Fu et al. [57] add semantically-similar tokens into the green list; and Ren et al. [159] use a trained MLP to generate semantic values. However, these approaches do not control the probability associated with green-list membership due to their heuristic designs and therefore lack formal Type-I error guarantees. In comparison, our work introduces speculative decoding [107] to the area of statistical watermarking and uses a proposal LLM to capture semantics explicitly, thus enjoying provable Type-I error guarantees.

While statistical watermarking commonly uses private keys for generation and detection,

watermarks can also be injected with private forgeability and public verifiability [49, 119], hence functioning effectively as digital signatures.

Despite its formal clarity, watermarking faces threats from various attack algorithms in practice [99, 100, 102, 165, 228]. Notably, due to several attack methods that can destroy watermarks while preserving quality, tamper resistance, or robustness becomes an important consideration. The result by Zhang et al. [228] asserts that tamper resistance is only feasible against a well-specified class of attacks, rather than against arbitrary attacks. To address the tamper resistance challenge, several works [37, 62] design robust pseudo-random codes as the basis for constructing robust watermarks. However, these methods remain largely theoretical and have yet to see practical implementation. Zhao et al. [235] design a new LLM decoding method and a tailored watermarking scheme for this decoding method, which improves tamper resistance. To support the long-term advancement of watermarking techniques, benchmark efforts [133, 152] are crucial in evaluating quality, efficiency, and tamper resistance of practical watermarking methods.

E-values and sequential hypothesis testing. The field of sequential hypothesis testing aims to design statistical methods that can accommodate streams of data and data-dependent stopping times [21, 161, 194]. The earliest work in the field relied on likelihood ratios and p-values. These tools are appropriate for a limited range of problems but in general fail to control Type-I error under data-dependent stopping rules and composite hypotheses. These deficiencies led to the development of tests based on nonnegative supermartingales, which yield e-values at arbitrary stopping times under composite null hypotheses [68, 158, 170, 192]. Moreover, this framework also provides tools for analyzing and improving statistical power. For example, Grünwald et al. [68] introduced worst-case growth rate optimality (GROW) as a criterion to assess power for a composite alternative. More recently, Waudby-Smith et al. [202] established optimal rates for a broad class of betting-based e-processes (a generalization of nonnegative supermartingales), and also obtained matching upper and lower bounds on mean rejection times based on these processes.

1.3 Notation

We summarize common notation that is used repeatedly across chapters. Chapter-local symbols are defined where they are used.

For a set A, we write A^c as the complement of A, $|A|$ as its cardinality, and $2^A := \{B : B \subseteq A\}$ as its power set. We use $(x)_+ = \max\{x, 0\}$, $x \wedge y = \min\{x, y\}$, and $x \vee y = \max\{x, y\}$. The asymptotic notations $O(\cdot)$, $\Omega(\cdot)$, and $\Theta(\cdot)$ hide absolute positive constants. We write $\mathbb{P}$ and $\mathbb{E}$ for probability and expectation. Given a sample space $\mathcal{X}$, $\Delta(\mathcal{X})$ denotes the set of all probability measures over $\mathcal{X}$, and $\mathrm{supp}(\mu)$ denotes the support of the distribution μ.

Chapter 2

Sample Complexity and Expressiveness of Test-Time Scaling Paradigms

This chapter first establishes 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, this chapter presents an expressiveness result for the self-correction approach with verifier feedback: it enables generic Transformer architectures to simulate online learning over a pool of experts at test time, thus achieving multi-task solving without prior knowledge of the specific task associated with a user query or explicit identification of the best expert. This result extends the representation theory of Transformers from single-task to multi-task settings. Finally, this chapter empirically validates the theoretical results, and demonstrates the practical effectiveness of self-correction methods.

2.1 Introduction

Over the past several years, Large Language Models (LLMs) have witnessed remarkable advances, achieving unprecedented performance across a broad spectrum of applications [1, 23, 24]. Driven by the paradigm of chain-of-thought (CoT) reasoning [204], 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 of LLMs—a phenomenon known as test-time scaling [22, 45, 143, 209]. Various methods have been proposed to effectively utilize additional test-time compute, including self-consistency [22, 33, 138, 198], best-of-n sampling [85, 135, 154, 168, 181], Monte Carlo Tree Search (MCTS) [31, 58, 118, 186, 195, 232], and self-correction [36, 66, 103, 127, 206, 231]. Powered by test-time scaling paradigms, several reasoning models, such as OpenAI-o1 [144] and DeepSeek-R1 [45], have achieved remarkable

success in many complex tasks [40, 71, 84, 173, 227].

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 [50, 90, 114, 131], two fundamental challenges remain unresolved:

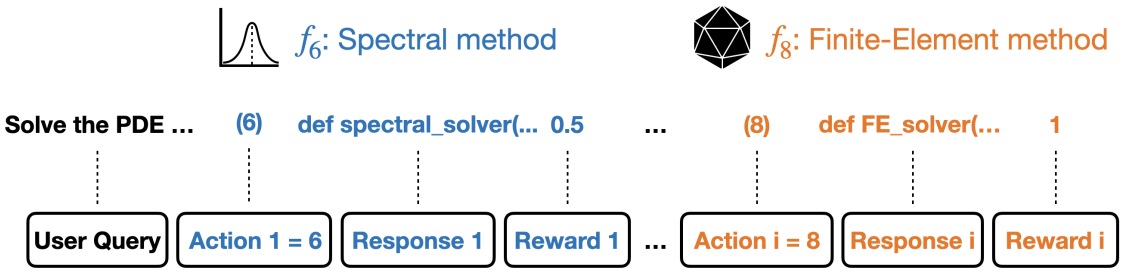


Figure 2.1: An illustration of test-time online learning (figure adapted from [110]), where the Transformer progressively learns that finite-element method solves the partial differential equation with higher accuracy.

1. Many test-time scaling approaches rely on repeated sampling from the same LLM to select a final answer [22, 33, 85, 97, 135, 138, 154, 168, 181, 198, 210]. 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 [46, 223], while their ability to express multiple tasks at the same time is under-studied. In contrast, practical LLMs have shown remarkable success in adapting to new, diverse tasks at test time via self-correction [36, 66, 69, 103, 127, 206, 231] or In-Context Reinforcement Learning [134, 141, 180, 187]. This gap underscores a need to understand how Transformers express multiple task-solving capabilities inherently and learn to adapt their behavior to the right task via test-time interaction. Therefore, 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 chapter 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 2.1.1 (Informal statement of Theorem 2.3.1 and Theorem 2.3.2)**.** *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 2.1: 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., negative 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.

Proposition 2.1.2 (Informal statement of Theorem 2.4.7)**.** *There exists a generic way to construct a wider transformer f from any Transformer-based expert models $f_1, \ldots, f_K$ such that, for any task solvable by the base experts, f can solve this task by self-correction which generates a sequence of responses with regret $o(1)$.*

Proposition 2.1.2 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 2.1.2 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 these results through controlled experiments.

2.2 Preliminaries

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

Definition 2.2.1 (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 $i = 1, \ldots, n$, $X_i^{(1)} = \mathrm{pe}(\theta(v_i); v_1, \ldots, v_i)$.
2. For each $l = 1, \ldots, L-1$, compute each $X_i^{(l+1)}$ for $i = 1, \ldots, n$ by

$$X_i^{(l+1)} = \sum_{h=1}^{H} \sum_{j=1}^{i} \frac{\exp\left(s_h^{(l)}(X_i, X_j)\right)}{Z_h^{(l)}} \cdot \mathbf{V}_h^{(l)} X_j^{(l)}, \tag{2.1}$$

   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

$$p_f(y|v_1, \ldots, v_n) = \mathrm{Softmax}(\vartheta(y)^\top X_n^{(L)}),\ y \in \mathcal{V}.$$

In particular, we assume the softmax attention layer has precision ϵ in the sense that: 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. (2.1).

While classical positional encoders are solely dependent on the index of the current token (i.e. we may write $\mathrm{pe}(\theta(v_i); v_1, \ldots, v_i) = \mathrm{pe}(\theta(v_i); i)$), recent advances [61, 70, 229] have 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.

Definition 2.2.2 (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 $\mathcal{V}_1, \ldots, \mathcal{V}_K$ 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 $\mathcal{V}_1, \ldots, \mathcal{V}_K$, i.e.

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

Test-time scaling. In this chapter, 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.

2.3 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:

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

where $o^\star$ 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 [78]. Under this setting, we assume 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 2.3.1 (Sample Complexity of Self-Consistency)**.** *Let $O = |\mathcal{O}| < \infty$. There exist numerical constants $c, C > 0$ such that: when $n \geq \frac{C \log(O/\delta)}{\Delta^2}$, self-consistency with n i.i.d. samples is able to produce the correct answer with probability at least $1 - \delta$; when $n \leq \frac{c}{\Delta^2}$, there exists a hard instance where self-consistency with n i.i.d. samples fails to produce the correct answer with constant probability.*

[1]If multiple correct answers exist, the self-consistency result applies after canonicalizing answers into equivalence classes and assigning probability mass to these classes.

Theorem 2.3.2 (Sample Complexity of Best-of-n)**.** *Assume $r(o^\star) > r(o), \forall o \neq o^\star$. There exist numerical constants $c, C > 0$ such that: when $n \geq \frac{C \log(1/\delta)}{\Delta}$, best-of-$n$ with n samples produces the correct answer with probability at least $1 - \delta$; When $n \leq \frac{c}{\Delta}$, 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 establish 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 has an advantage, requiring only $\Theta(1/\Delta)$ 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 [184, 209].

2.4 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.

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 2.4.1 (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.

2.4.1 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.

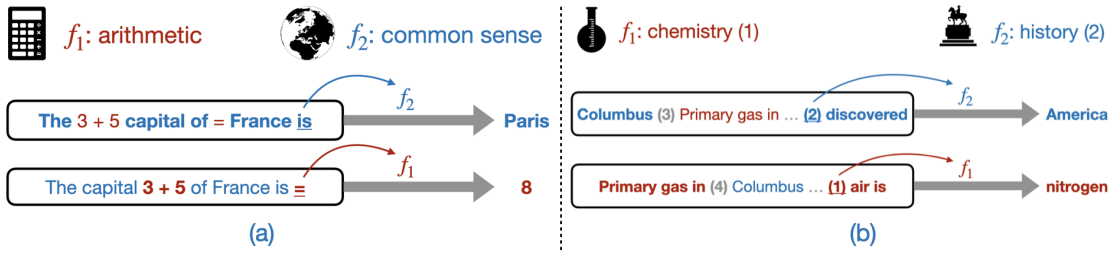


Figure 2.2: (a): Illustration of Proposition 2.4.2. In the first query, f_2 is called to solve the common sense problem by attending to only blue tokens. In the second query, f_1 is called to solve the arithmetic problem by attending to only red tokens. (b): Illustration of Proposition 2.4.4. In the 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 2.4.2 (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*

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

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 2.4.3. 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 2.4.4 (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*

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

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

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

then we have

$$p_{\widetilde{f}}(\cdot|v) = p_{f_\kappa}(\cdot|u) \tag{2.2}$$

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 2.4.5. 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 $\log(N_{\max})$.

2.4.2 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 2.4.6 (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 $\max_{a^* \in \mathcal{A}} r(a^*) - \mathbb{E}[r(a_T)] \le \mathrm{reg}(T) = o(1)$ where $a_1, \ldots, a_T$ are generated in the following way:

$$a_t \sim p_f(\cdot \mid a_1, R_1, \ldots, a_{t-1}, R_{t-1}),\ \forall t = 1, \ldots, T,$$
$$\mathbb{E}[R_t \mid a_t, \mathcal{H}_{t-1}] = r(a_t),\ \forall t = 1, \ldots, T.$$

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 [26, 48, 86]. 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 [50, 114, 126, 223]: For example, [117] provide constructions to approximate upper confidence bounds and Thompson sampling algorithms with average regret $O(1/\sqrt{T})$. Consequently, their construction is not the primary focus of this chapter.

Algorithm 2.4.1 Self-correction with verifier

procedure GENERATION(q) ▷ $q = q_1 \dots q_{n_0}$ denotes the user query.
  prompt $\leftarrow q$
  for $t = 1, \dots, T$ **do**
    $a^{(t)} \sim p_{\widetilde{f}}(\cdot \mid \text{prompt})$ ▷ $a^{(t)}$ designates which expert to use in t-th iteration
    prompt $\leftarrow$ prompt$|a^{(t)}$ ▷ Update the prompt autoregressively, $|$ represents token concatenation.
    for $i = 1, \dots$ **do**
      $u_i^{(t)} \sim p_{\widetilde{f}}(\cdot \mid \text{prompt})$ ▷ Generate t-th response autoregressively
      prompt $\leftarrow$ prompt$|u_i^{(t)}$ ▷ Update the prompt autoregressively
      if $u_i^{(t)} = \text{EOS}$ **then**
        Break
    $r^{(t)} \leftarrow r(q, u^{(t)})$, prompt $\leftarrow$ prompt$|r^{(t)}$ ▷ Query verifier to obtain reward of t-th response
  Return $u^{(T)}$

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 2.4.1, 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 2.4.7 (Regret Minimization via Self-Correction)**.** *For any $H, L, K, N_{\max} \in \mathbb{Z}_+$, task set $\mathcal{Q}_T$, token spaces $\mathcal{V}_0, \mathcal{V}_1, \dots, \mathcal{V}_K, \mathcal{A}$ ($|\mathcal{A}| = K$) 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,*

- *K **expert Transformers**: $f_k = (\theta_k, \mathrm{pe}_k, (\mathbf{K}_{k;h}^{(l)}, \mathbf{Q}_{k;h}^{(l)}, \mathbf{V}_{k;h}^{(l)})_{h\in[H],l\in[L]}, \vartheta_k, \mathcal{V}_k)$ for $k \in [K]$, such that for any task query $q \in \mathcal{Q}_T$, there exists one expert f_{k^*} achieving 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**: $f_0 = (\theta, \mathrm{pe}, \mathbf{K}_{0;h}^{(l)}, \mathbf{Q}_{0;h}^{(l)}, \mathbf{V}_{0;h}^{(l)})_{h\in[H],l\in[L]}, \vartheta, \mathcal{V}_0 \cup \mathcal{A})$ 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 $v \in \mathcal{Q}_T$, if the response sequence $u^{(1)}, \ldots, u^{(T)}$ generated by the protocol in Algorithm 2.4.1 has total length $\leq N_{\max}$, then we have

$$\max_{u^*\in\mathcal{V}^\omega} r(v,u^*) - \mathbb{E}[r(v,u^{(T)})] \leq \lambda + \mathrm{reg}(T)$$

Remark 2.4.8. While the general-purpose Transformer ϕ can be applied to construct the brute-force Transformer $\widetilde{f}$ that simply tries every expert, we note that the generality of Definition 2.4.6 allows us to construct more powerful Transformers beyond brute-force search. Leveraging the structures in the problem and the expert pool, it is entirely possible to identify the correct expert using $\ll K$ trials [55, 162].

As a consequence of Theorem 2.4.7, the transformer $\widetilde{f}$ can solve K distinct tasks at inference time, without requiring prior knowledge of the specific task or identification of the best expert. The architecture-level composition is *general-purpose* in the sense that it does not depend on the internal parameters of the expert Transformers. The regret guarantee, however, depends on the assumed verifier feedback model and on the existence of a regret-minimization Transformer for the induced reward function. To our knowledge, this is among the first theoretical expressiveness analyses of Transformer self-correction in this multi-expert setting. Furthermore, our theory aligns with the empirical finding that LLMs can progressively optimize outcome rewards during test-time [134, 155, 180, 187].

2.5 Experiments

In this section, we conduct synthetic experiments to show that Transformers can self-correct with verifier feedback.[2]

[2]Code: `https://github.com/LithiumDA/Representation-Ability-of-Test-time-Scaling`.

2.5.1 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-turn generation is challenging. In this case, self-correction can play a critical role if Transformers have this capability. 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 {`a`, `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 encourages models to fail on their first attempt, thereby eliciting 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 [98] for 5 epochs. More implementation details can be found in Appendix A.2.

Results. Test set accuracy across different inference settings is shown in Figure 2.3. 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.

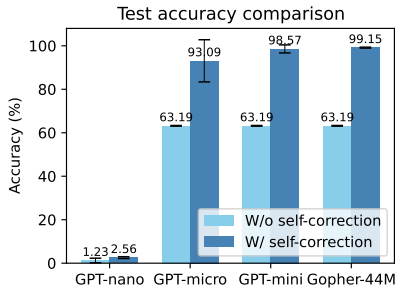


Figure 2.3: Accuracy comparisons of different models with/without self-correction at test time.

2.5.2 Evaluation of sample complexity

Dataset. We conduct experiments on the AIME 2024 & 2025 datasets [129], 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` [212], and `Llama-3.2-3B-Instruct` [124], 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-n and self-correction).

Results. We compare the accuracy of self-consistency, best-of-n, and self-correction under different sample sizes. Notably, as summarized in Table 2.1, best-of-n 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 demonstrate that modern Transformer models are sufficiently expressive to implement self-correction mechanisms when combined with verifiers, thus consistent with our theoretical results in Section 2.3 and 2.4.

2.6 Conclusion

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

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

to multi-task settings. Our experiments validate the theoretical separation and confirm that it requires additional model capacities for Transformer to implement self-correction.

Chapter 3

Bits-to-Rounds Sampling for Diffusion Language Model Inference

This chapter demonstrates the information-theoretic bottleneck of scaling up the inference speed of Diffusion Language Models (DLMs) through a bits-to-rounds principle that lower bounds the number of decoding rounds by the sample's total information and the confidence threshold serving as a per-round information budget. Motivated by this theory, this chapter proposes Explore-Then-Exploit (ETE), a training-free decoding strategy that maximizes information throughput and decoding efficiency. Experiments across diverse benchmarks verify the theoretical bounds and demonstrate that ETE consistently reduces the number of decoding rounds compared to confidence-only baselines without compromising generation quality. Furthermore, ETE integrates efficiently with KV caching, translating these algorithmic gains into improved tokens-per-second throughput.

3.1 Introduction

Diffusion Language Models (DLMs) exploit bidirectional attention to decode multiple tokens simultaneously (i.e., parallel decoding). At the start of generation, all output tokens are masked. In each decoding round, DLMs iteratively unmask a set of potentially non-sequential tokens until the entire output sequence has been uncovered. This ability to generate multiple tokens per forward pass yields substantial throughput gains compared to the inherently sequential decoding process of standard autoregressive models. Many popular diffusion models employ a hybrid decoding strategy in which the output sequence is divided into equal-sized blocks, and each block undergoes the diffusion unmasking process sequentially. Prior works have found this block-based decoding approach to deliver the best tradeoff between speed and accuracy when compared with full parallel decoding and autoregressive decoding.

Empowered by Masked Diffusion Model (MDM) architectures, recent works [139, 153, 239] have demonstrated strong scalability and competitive performance of DLMs across a broad

range of tasks. Moreover, commercial DLMs [44, 105, 224] have made significant industrial impact through superior inference throughput and speed-quality tradeoffs.

Despite these successes, DLM inference faces a fundamental challenge: parallel decoding incurs inherent performance degradation. Independently sampling tokens from marginal conditional distributions of DLMs does not reproduce the true joint distribution [51, 207]. A widely adopted remedy is to unmask only high-confidence tokens at each iteration [27, 63, 139, 207]. These confidence-based heuristics partially mitigate the tension between decoding accuracy and throughput by approximating the true joint distribution. However from an information-theoretic viewpoint, high probability implies low information, thus prioritizing high-confidence tokens reduces the information revealed per step. Given that the output $\mathbf{x}$ has fixed total information content $-\log p(\mathbf{x})$, decoding strategies that prioritize only the highest-confidence tokens should require more iterations than approaches that do not depend solely on this heuristic when unmasking tokens. This argument suggests a relationship between the computational cost (number of decoding rounds) and total information $-\log p(\mathbf{x})$, thus motivating our first question:

Can the computational cost of decoding be characterized in terms of fundamental information-theoretic quantities?

If such a characterization exists, it would naturally suggest a design principle: *to minimize decoding rounds, we should maximize the information decoded per round*. However, current approaches fail to follow this principle due to two complementary limitations. First, confidence-based methods prioritize high-confidence tokens and inevitably unmask low-confidence tokens only when there are no high-confidence ones available. Second, existing block-based decoding strategies process blocks strictly sequentially. This causes the final few tokens in each block to be decoded essentially one by one rather than simultaneously with tokens in future blocks, thereby limiting the *number of tokens* that can be decoded per round. This gap motivates our second question:

How can we systematically maximize the information decoded per round, both by identifying high-information tokens and by expanding parallel decoding opportunities?

In this chapter, we formally address the above questions. Our main contributions are twofold:

Information-theoretic lower bound. We formalize the relationship between computational cost of confidence-based heuristics and information content by proving a lower bound on the decoding rounds:

$$R \geq \max\left(\frac{-\log p(\mathbf{x}) - \epsilon}{f}, \frac{-\log p(\mathbf{x})}{\log\left(\frac{n+1}{(1-f)n+1}\right)}\right),$$

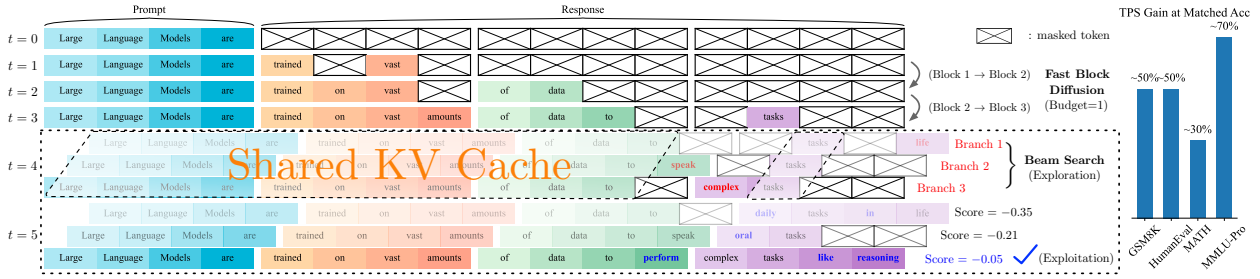


Figure 3.1: Illustration of **E**xplore-**T**hen-**E**xploit (ETE): Fast block diffusion scheme proceeds by sequentially unlocking a new block to the right and applying confidence-based parallel sampling to all past blocks. At $t = 4$, targeted exploration is triggered to apply beam search to 3 candidate exploration tokens (in red) with a shared KV cache of the previous unmasked positions. At the next step, several high-confidence tokens (in blue) are unlocked, and branch 3 with the highest score is committed. We summarize the overall TPS (tokens per second) gain at matched accuracy across benchmarks, with detailed experimental results in Section 3.5.

where n is the sequence length, ϵ is the total approximation error of parallel decoding, and f is the confidence threshold factor. This bound reveals a fundamental structure:

$$\text{Rounds} \geq \frac{\text{Total Information}}{\text{Information per Round}}.$$

Reducing confidence threshold f (stricter requirements) shrinks the denominator and a smaller approximation error ϵ increases the numerator, forcing more decoding rounds.

Exploration-aware decoding algorithm with cross-block decoding. Motivated by our theoretical insights and empirical observations, we introduce **E**xplore-**T**hen-**E**xploit (ETE), a new algorithmic framework that systematically maximizes information decoded per round through two complementary mechanisms: (i) *Fast block diffusion sampling*: We extend standard block diffusion by enabling cross-block parallel decoding; (ii) *Strategic exploration mechanisms*: We apply look-ahead search over informative positions to identify tokens that unlock the largest cascades of high-confidence decisions.

Together, these mechanisms directly amplify the denominator in our lower bound by increasing both the information per token and the number of tokens decoded per round. Across four standard benchmarks (MATH, GSM8K, HumanEval, MMLU-Pro), ETE consistently outperforms strong confidence-based baselines [207] in both decoding efficiency and output quality. We further identify a *free-lunch regime* where beam search with a shared KV cache introduces negligible wall-clock overhead, enabling exploration at minimal practical cost and even higher throughput (tokens per second). These results underscore the importance of principled exploration in diffusion decoding and demonstrate a promising path toward closing the gap between parallel DLM inference and the joint distribution it aims to approximate.

3.2 Background

Notation. We use bold symbols $\mathbf{x} = (x^1, \ldots, x^N)$ to represent a sequence of N tokens. We also use the standard big-O notation: $O(\cdot)$ and $\Omega(\cdot)$ to hide absolute positive constants. Let $p^i(\cdot)$ denote the marginal distribution at position i induced by p. For brevity, we omit the subscript and write $p(\cdot)$ when i is clear from context.

Masked diffusion models. Masked diffusion models (MDMs) are a subclass of discrete diffusion models. By extending the vocabulary $\mathcal{V}$ with a mask token [M], MDMs progressively transform a clean sequence $\mathbf{x}_0 = (x_0^1, \ldots, x_0^N)$ into a fully masked sequence $([\mathrm{M}], \ldots, [\mathrm{M}])$ in the forward process following an absorbing noising schedule:

$$q_{s|0}(\mathbf{x}_s \mid \mathbf{x}_0) = \prod_{i=1}^{N} q_{s|0}(x_s^i \mid x_0^i),$$

$$q_{s|0}(x_s^i \mid x_0^i) = \begin{cases} \alpha_t, & x_s^i = x_0^i, \\ 1 - \alpha_t, & x_s^i = [\mathrm{M}]. \end{cases}$$

At training time, a parametric model p_θ is trained to take $\mathbf{x}_s$ as input and predict all masked tokens simultaneously, by fitting the ELBO loss [139, 149, 164, 174]:

$$\mathcal{L}(\theta) = -\mathbb{E}_{s, x_0, x_s} \left[\frac{1}{s} \sum_{i=1}^{N} \mathbb{1}(x_s^i = [\mathrm{M}]) \log p_\theta(x_0^i \mid \mathbf{x}_s) \right].$$

Moreover, this ELBO loss can be reformulated into a time-agnostic version [236]:

$$x^j_{\sigma(<i)} = \begin{cases} x_0^j, & j \in \sigma(< i) \\ [\mathrm{M}], & j \in \sigma(\geq i) \end{cases},$$

$$\mathcal{L}(\theta) = -\mathbb{E}_{\sigma \sim \mathrm{Unif}(S_N)} \left[\sum_{i=1}^{N} \sum_{k \in \sigma(\geq i)} \frac{1}{N - i + 1} \log p_\theta(x_0^k | \mathbf{x}_{\sigma(<i)}) \right]$$

where σ is a random permutation uniformly sampled over all permutations on $[N]$, and $\mathbf{x}_{\sigma(<i)}$ represents a partially masked version of the clean sequence $\mathbf{x}_0$, where indices in $\sigma(\geq i)$ are masked. This demonstrates that p_θ intrinsically parametrizes an *any-order autoregressive model*, able to predict the entire sequence with any specified σ.

Inference of masked diffusion models. At inference time, MDMs discretize the reverse process by iteratively reconstructing clean sequences from a fully masked state. Specifically, we start with a fully masked sequence $\mathbf{x}_0 = [[\mathrm{M}]^{\otimes N}]$ and randomly sample a generation order $\sigma \sim \mathrm{Unif}(S_N)$. At step $t \in [N]$, we unmask position $\sigma(t)$ by sampling from $x^{\sigma(t)} \sim p(x^{\sigma(t)} | \mathbf{x}_{t-1})$ and let $\mathbf{x}_t$ be $\mathbf{x}_{t-1}$ with position $\sigma(t)$ replaced by $x^{\sigma(t)}$. In this process, a token remains fixed for the remainder of the denoising steps once it is unmasked, and this token-by-token random-order sampler is theoretically consistent with the ELBO reformulation [149].

Block diffusion sampling. Given a prompt sequence $\mathbf{x}_{\text{prompt}}$ of length n_0 and a target generation length n, the sequence is initialized as $\mathbf{x}_0 = [\mathbf{x}_{\text{prompt}}, [\text{M}]^{\otimes n}] \in \mathcal{V}^{n+n_0}$, which concatenates the prompt (as a prefix) with a response window of n mask tokens. Given a block length n_b, the response window is partitioned into $L = n/n_b$ consecutive blocks $\mathcal{B}_b = [n_0 + (b-1)n_b : n_0 + bn_b]$ $(b = 1, \ldots, L)$, each of length n_b. Tokens are then decoded block-by-block from left to right. This approach supports flexible-length generation and improves inference efficiency with KV caching [14]. We provide its visualization in Panel (a) of Figure B.1 in Appendix B.2.

Confidence-based parallel decoding. Parallel decoding aims to unmask multiple tokens simultaneously in a single inference step. At each decoding step t within block b, given the partially unmasked sequence $\mathbf{x}_t$, the model first generates predictions at all masked positions in the block via greedy parallel decoding of their conditional marginals:

$$\hat{x}^i_{t+1} = \arg\max_{v \in \mathcal{V}} p^i_\theta(v \mid \mathbf{x}_t),$$
$$i \in S := \{i : x^i_t = [\text{M}] \text{ and index } i \text{ is in block } b\}$$

Since the product of conditional marginal distributions over all masked tokens may differ substantially from the true conditional joint distribution, parallel-decoding strategies selectively commit only a subset of predictions to ensure small approximation error. As the predominant strategy in parallel decoding, confidence-based parallel-decoding methods selectively unmask tokens in block b based on their confidence scores: $c^i = p^i_\theta(\hat{x}^i_{t+1} \mid \mathbf{x}_t)$. We describe three variants of confidence-based decoding strategies:

(1) **Fixed-number scheme:** Given a fixed number $k > 0$ we expect to decode per round, Nie et al. [139] unmask the top-k confident tokens within the block.

(2) **Static confidence threshold:** Yu et al. [222] propose a static threshold scheme that unmasks all tokens exceeding a fixed confidence threshold $C \in (0, 1)$.

(3) **Dynamic confidence threshold:** Wu et al. [207] extend the static threshold strategy to a dynamic variant that sorts confidences as $c^{(1)} > \cdots > c^{(m)}$ and determines the maximum integer k satisfying $(k+1)(1-c^{(k)}) < f$, where $f > 0$ is a predefined threshold. The algorithm then unmasks the corresponding top-k confident tokens $\hat{x}^{(1)}_{t+1}, \ldots, \hat{x}^{(k)}_{t+1}$.

Regardless of which specific selection strategy is used, we refer to this step as an *exploitation* step, as it prioritizes and exploits high-confidence predictions. Intuitively, tokens are nearly *conditionally independent* when they are highly confident. Notably, for the latter two confidence-based decoding strategies, decoding would stall when the current block contains no tokens exceeding the confidence threshold. To prevent this, the highest-confidence token is unmasked regardless of whether its confidence exceeds the threshold. We refer to this as an *implicit exploration* step to distinguish it from the *exploitation* step, since an implicitly

low-confidence token is decoded in this step. We provide the pseudocode for the entire confidence-based decoding process with a static confidence threshold C in Algorithm B.3.1 in Appendix B.3.1.

3.3 Inefficiency of Confidence-based Decoding

While exploitation steps form the core of current confidence-based decoding, they face fundamental limitations that need investigation. Before formalizing our theoretical framework, we present an empirical study to quantify the inefficiency of parallel decoding.

3.3.1 Information contribution of exploration and exploitation

We evaluate the dynamic confidence-based decoding strategy [207] across different confidence factors on the GSM8K dataset [40], with the detailed experimental setup deferred to Section 3.5.1. Figure 3.2 shows how much information is revealed by different decoding steps in confidence-based parallel decoding. Decoding rounds are divided into *exploitation*, which commits high-confidence tokens, and *(implicit) exploration*, which commits low-confidence tokens when no confident tokens are available. We observe that exploration is significantly more *information-efficient* than exploitation across all confidence factors f. For example, at $f = 0.4$, exploration uses only 23.5% of rounds but contributes 66.7% of total information, while exploitation consumes 76.5% of rounds yet contributes only 33.3%. This gap persists across all settings: exploration achieves efficiency ratios around 3, whereas exploitation remains below 1.

We also provide illustrative examples in structured modalities such as code and profiles (in Appendix B.2.1), where greedily prioritizing high-confidence positions is globally inefficient. In contrast, decoding low-confidence tokens leads to faster generation since they carry substantial information that constrains many downstream positions.

This observation motivates two directions: (i) theoretically analyzing the fundamental limits of confidence-based decoding algorithms, and (ii) empirically designing principled exploration strategies that identify and target high-information yet low-confidence tokens.

3.3.2 Information-theoretic lower bounds on decoding rounds of confidence-based algorithms

Setup and notation Let $p(\cdot)$ be a distribution over a length-n sequence that models the conditional probability $p(x^A|x^B)$ for any disjoint subset A, B of $[n] \equiv \{1, \ldots, n\}$. A *parallel decoding schedule* is an ordered partition of indices

$$A_1 \cup A_2 \cup \cdots \cup A_R = [n], \quad A_i \cap A_j = \emptyset$$

such that x^{A_r} is the set of tokens unmasked in rounds r for $r = 1, \ldots, R$. At the start of round r, the set of unmasked indices is $C_r \triangleq \bigcup_{s<r} A_s$, and the masked indices are $U_r \triangleq [n] \setminus C_r$. A

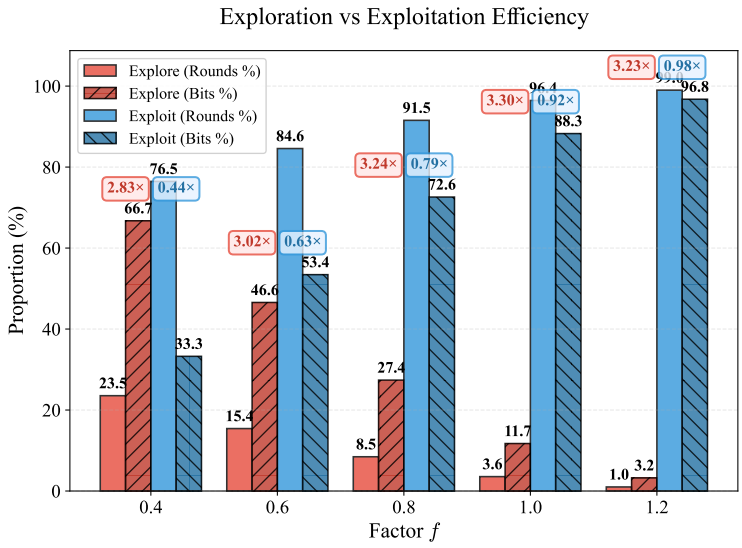


Figure 3.2: Information efficiency of exploration versus exploitation in confidence-based parallel decoding. For each confidence factor f, we report the fraction of decoding rounds and the fraction of total information (bits) contributed by (implicit) exploration and exploitation. Bits are measured as accumulated conditional negative log-probabilities ($-\log p(\mathbf{x}_{t+1}|\mathbf{x}_t)$). Each group shows four bars: exploration rounds (%), exploration bits (%), exploitation rounds (%), and exploitation bits (%), averaged over 200 GSM8K samples. Colored boxes indicate the efficiency ratio (bits % / rounds %) for exploration (red) and exploitation (blue).

parallel decoding algorithm selects the subset $A_r \subseteq U_r$ to be decoded in this round based on the conditional marginals $p(x^i|x^{C_r})$ over all masked indices $i \in U_r$. Due to parallel decoding, an *approximation error* will arise from the gap between the true conditional probability $p(x^{A_r} \mid x^{C_r})$ and the actual sampling probability $\prod_{i\in A_r} p(x^i \mid x^{C_r})$, and the error will accumulate with the decoding process. Define the *total approximation error* by

$$\epsilon = \ -\log p(\mathbf{x}) - \sum_{r=1}^{R} \sum_{i\in A_r} \left[-\log p(x^i \mid x^{C_r})\right] \ . \tag{3.1}$$

In confidence-based methods, token x^i is unmasked if and only if its confidence $p(x^i \mid x^{C_r})$ satisfies a lower bound. Therefore, we adopt the following assumption that captures a broad family of practical confidence-based rules used in DLM decoding (e.g., [207]).

Assumption 3.3.1 (Dynamic threshold confidence decoding)**.** The selection rule for each round r ensures that every chosen position $i \in A_r$ satisfies

$$(1 + |A_r|)(1 - p(x^i \mid x^{C_r})) \ \leq f, \quad f \leq 1.$$

The quantity $1-p(x^i \mid x^{C_r})$ measures the algorithm's uncertainty regarding position i, and the factor $(1 + |A_r|)$ enforces that as more tokens are decoded jointly in round r, each token

must individually meet a stricter confidence threshold. This assumption also theoretically guarantees that *greedy parallel decoding yields the same result as greedy sequential decoding* on A_r (Theorem 1, Wu et al. [207]). Under this assumption, we prove an explicit lower bound on the number of rounds required by *any* decoding algorithm obeying this dynamic confidence constraint.

Theorem 3.3.2 (Step lower bound for confidence-based parallel decoding)**.** *Consider any parallel decoding schedule and any sequence* $\mathbf{x} = (x^1, \ldots, x^n)$ *satisfying Assumption 3.3.1, and let the total approximation error* ϵ *be defined in Eq.* (3.1)*. Then the number of rounds* R *must satisfy*

$$R \geq \max\left(\frac{-\log p(\mathbf{x}) - \epsilon}{f}, \frac{-\log p(\mathbf{x})}{\log\left(\frac{n+1}{(1-f)n+1}\right)}\right). \tag{3.2}$$

Interpretation. Theorem 3.3.2 formalizes a fundamental tradeoff in confidence-based parallel decoding. The lower bound explicitly couples the required number of rounds R to two quantities: it is proportional to the total amount of information in the sequence $-\log p(\mathbf{x}) - \epsilon$ and inversely proportional to the per-round information limit f. Furthermore, we have a conservative bound regardless of the approximation error ϵ. This result reveals that confidence-based heuristics become inherently inefficient in either high-entropy (large $-\log p(\mathbf{x})$) or strict decoding (small f) regime. Thus, this structural inefficiency motivates us to *actively identify* and *resolve high-information positions* to overcome the fundamental limitation. The proof is provided in Appendix B.1.

3.4 ETE: Explore-Then-Exploit

3.4.1 From theory to algorithm

Our information-theoretic analysis in Section 3.3 reveals a simple but fundamental relationship:

$$\text{Rounds} \geq \frac{\text{Total Information (Bits)}}{\text{Information per Round (Bits per Round)}}. \tag{3.3}$$

Motivated by this inequality, we propose two approaches to improve the denominator.

- **Expanding the decoding canvas (Fast block diffusion sampling):** By enabling parallel decoding across multiple blocks simultaneously, we increase the *number of tokens* that can be decoded in each round, thereby improving the opportunities for high-information contribution each round.
- **Targeting high-information tokens (Strategic exploration):** By explicitly identifying and decoding high-entropy tokens–those that carry the most information and unlock cascades of downstream predictions–we increase the *information per token* decoded in each round.

In the remainder of this section, we detail each component and show how they combine to form ETE (**E**xplore-**T**hen-**E**xploit), a principled algorithm that operationalizes these information-theoretic principles.

3.4.2 Fast block diffusion sampling

Standard block diffusion processes blocks sequentially, unlocking block $b+1$ only after block b is completely decoded. This rigid left-to-right constraint artificially limits parallelism and under-utilizes the bidirectional attention mechanism. Following block diffusion LLaDA sampling [139, 207], ETE partitions the sequence $\mathbf{x}$ of length n into L contiguous blocks $\mathcal{B}_1, \ldots, \mathcal{B}_L$, with each block's length being $n_b = n/L$. Crucially, ETE introduces two key modifications:

1. *Budget-based progression*: Rather than waiting for complete block decoding, we assign a uniform sampling budget of N decoding rounds per block and unlock the next block once this budget is exhausted. This prevents the algorithm from stalling in low-throughput regimes where blocks are nearly complete and few tokens remain to be decoded.

2. *Inter-block parallel decoding*: Unlike prior methods that restrict decoding to the current block, ETE allows high-confidence tokens in *earlier blocks* to be unmasked in parallel with the current block. This bidirectional information flow enables future positions to inform and confirm earlier predictions.

Formally, let b_t be the block index at step t and $\mathcal{M}_t$ be the set of all masked positions after step $t-1$. The feasible positions for unmasking at step t are:

$$S_t = \mathcal{M}_t \cap \left(\cup_{b=1}^{b_t} \mathcal{B}_b\right), \tag{3.4}$$

which includes all masked positions in the current block and all previous blocks. After exhausting the budget for the last block, we apply a few additional decoding rounds to ensure all positions are decoded. A visualization of our approach is provided in Panel (b) of Figure B.1 in Appendix B.2.

By enabling cross-block parallel decoding, fast block diffusion increases the *feasible set* of tokens that can be decoded in each round. High-confidence tokens that emerge in earlier blocks during later-block processing can now be immediately committed, extracting more information per inference step. This directly expands the denominator ("bits per round") in our theoretical framework.

3.4.3 Confidence-based exploitation

ETE retains the benefits of confidence-based token selection for exploitation. Specifically, let $\mathbf{x}_t$ be the full sequence at step t. We run the diffusion LM once to obtain marginal probabilities $p_\theta^i(\cdot \mid \mathbf{x}_t)$ for all positions $i \in S_t$. Following Yu et al. [222], ETE performs greedy decoding $\hat{x}_{t+1}^i = \arg\max_{v \in \mathcal{V}} p_\theta^i(v \mid \mathbf{x}_t)$ and commits exploitation tokens whose confidence exceeds the threshold $C > 0$:

$$x_{t+1}^i = \begin{cases} \hat{x}_{t+1}^i, & \text{if } c^i(\mathbf{x}_t) > C \text{ and } i \in S_t, \\ x_t^i, & \text{otherwise.} \end{cases} \tag{3.5}$$

where $c^i(\mathbf{x}_t) = p_\theta^i(\hat{x}_{t+1}^i | \mathbf{x}_t)$ is the confidence at index i given $\mathbf{x}_t$.

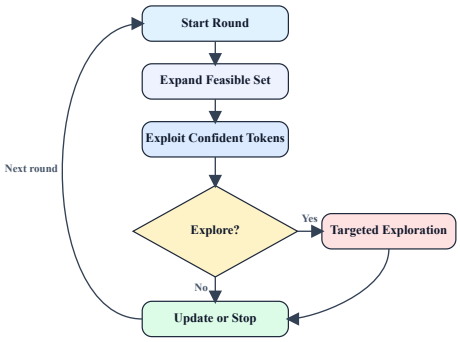


Figure 3.3: Minimal per-round workflow of Explore-Then-Exploit (ETE). At each round, ETE expands the feasible decoding set over the current and previously unlocked blocks, exploits high-confidence positions, and conditionally invokes targeted exploration when the frontier is information-poor.

3.4.4 Information-aware exploration

While fast block diffusion expands *how many* tokens can be decoded per round, it does not address *which* tokens carry the most information. Our empirical observations (Figure 3.2) and illustrative examples (in Appendix B.2.1) demonstrate that decoding some low-confidence tokens triggers cascades of newly confident predictions. However, confidence-based methods systematically avoid these high-information positions. To maximize bits per round, we must strategically target and decode these tokens through principled exploration.

ETE employs two complementary exploration mechanisms operating at different scales:

Inter-block implicit exploration. When a *previous block* lacks high-confidence tokens (none exceed threshold C), ETE unmasks the highest-confidence masked token in that block. This ensures forward progress in every active block, preventing decoding from stalling.

Targeted exploration via beam search. Beyond implicit exploration, ETE performs principled exploration within the *current block* by strategically identifying and testing high-information tokens through look-ahead beam search.

When to explore. ETE triggers targeted exploration when:

1. The average confidence in the decoding frontier falls below threshold γ, indicating an information-poor region where exploitation alone may require many rounds.

2. The block has sufficient remaining masked tokens ($> N_e$) to justify the exploration overhead.

This information-aware criterion ensures exploration focuses on genuinely difficult situations where the potential information gain justifies the computational cost.

What to explore. Rather than exploring uniformly, we target *medium-confidence* positions near a confidence level c^{info}. These positions represent genuine ambiguity (unlike near-certain tokens) yet have sufficient signal to resolve accurately (unlike tokens with near-zero confidence). Given beam size k, ETE identifies exploration candidates via:

$$\mathcal{H} = \text{Topk}_{i \in \text{current block} \cap \mathcal{M}_t} \Big(- c^i(\mathbf{x}_t) - c^{\text{info}} \ + \beta \cdot (i - (b_t - 1) \cdot n_b) \Big), \tag{3.6}$$

where the position bias $\beta > 0$ slightly favors later tokens to exploit bidirectional context.

Which to commit. For each candidate $j \in \mathcal{H}$, ETE creates a hypothesis $\mathbf{x}_{t+1;j}$ by fixing position j to $\hat{x}^j_{t+1}$ and performs *batched inference* on all $|\mathcal{H}| = k$ hypotheses simultaneously. Notably, we can leverage a *shared KV cache* on the already unmasked part of $\mathbf{x}_t$. The batched inference reveals how each exploration choice influences remaining masked positions. Each hypothesis is then evaluated via:

$$s(j) := \underbrace{\alpha \cdot \log c^j(\mathbf{x}_t)}_{\text{sample quality}} + \underbrace{\log \sum_{i \in S_{t+1} \setminus \{j\}} c^i(\mathbf{x}_{t+1;j}) \mathbf{1} \left(c^i(\mathbf{x}_{t+1;j}) \geq C \right)}_{\text{induced high-confidence tokens}}. \tag{3.7}$$

The first term ensures the exploration token has good sample quality, while the second term measures how many downstream tokens become high-confidence after committing this choice. The hyperparameter $\alpha > 0$ balances these two objectives. ETE identifies the candidate $j^\star = \arg\max_{j \in \mathcal{H}} s(j)$ achieving the highest score and commits the corresponding hypothesis sequence $\mathbf{x}_{t+1;j^\star}$. After this exploration, one exploitation step can be performed directly to decode these induced high-confidence tokens.

Committing high-entropy tokens collapses diffuse distributions, triggering confidence cascades through conditional dependencies, which can be exploited immediately. Critically, the batched inference introduces no additional model inference rounds: we reuse the outputs from the look-ahead beam search to commit the hypothesis sequence. Thus this simultaneously increases both the information per token and tokens per round, directly amplifying the denominator in our lower bound.

Integrating these components, we formalize the complete ETE sampling procedure in Algorithm B.3.2 with its detailed subroutines in Algorithms B.3.3-B.3.6 in Appendix B.3.2.

3.5 Experiments

3.5.1 Verification of Theorem 3.3.2

Experimental setup. We empirically validate the information-theoretic lower bound established in Theorem 3.3.2 using the GSM8K dataset [40]. Our experiments employ

`LLaDA-8B-Instruct` [139] as the base diffusion language model. We randomly sample 200 test questions and configure the model with a maximum generation length of 512 tokens and a block size of 64 tokens.

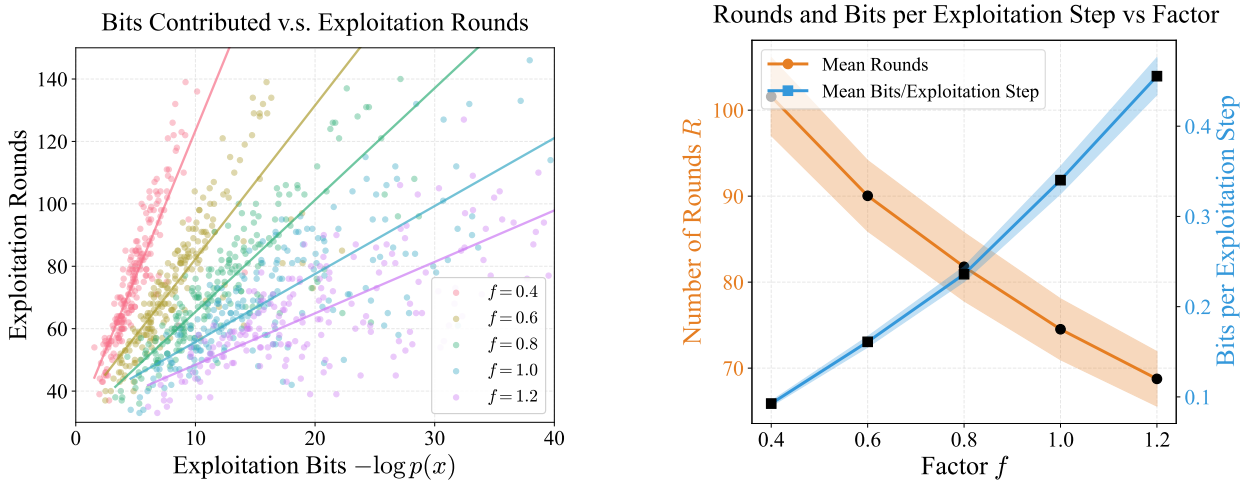
For each sample, we execute dynamic confidence-based parallel decoding with five different factors: $f \in \{0.4, 0.6, 0.8, 1.0, 1.2\}$[1]. To compute the true joint log-probability $-\log p(\mathbf{x})$ for each generated sequence, we accumulate the conditional log-probabilities by re-running the inference sequentially (token-by-token) following the left-to-right order when multiple tokens are decoded simultaneously in one round. We also evaluate the mathematical accuracy of the generated solution against the ground-truth answer, which is $78.5\%, 79.5\%, 78.5\%, 76\%, 76.5\%$ for $f = 0.4, 0.6, 0.8, 1.0, 1.2$, respectively.

Results and analysis. Panel (a) in Figure 3.4 presents the relationship between bits $(-\log p(\mathbf{x}))$ and the rounds (R) across different confidence factors f. When $f \leq 1$, the confidence factors exhibit clear linear relationships between bits and rounds, consistent with the lower bound in Equation (3.2). The strong linear correlations validate that the number of decoding steps scales linearly with sequence information content. Moreover, the slopes of the fitted lines decrease monotonically with the confidence factor f, confirming that stricter confidence requirements (smaller f) lead to greater step inefficiency. The near-linear relationship between f and the average bits per step also validates our theoretical per-round information budget. Panel (b) further demonstrates the relation between the per-step information decoded and confidence factor, establishing a nearly linear relationship between bits decoded per exploitation round and f, which substantiates our theory.

3.5.2 Computation overhead of beam search

We further investigate the computational overhead introduced by beam search under varying batch sizes. Figure 3.5 reports the average batched forward latency with KV caching for `LLaDA-8B-Instruct` as a function of beam size on A100, H100, H200, and B200 GPUs (we employ `Fast-DLLM` [207] implementation for KV caching). We observe a substantial "free lunch" region in which increasing the batch size incurs negligible additional cost: approximately 2 on H100, 4 on H200 and 7 on B200. Within this regime, the computational cost of exploring multiple candidate tokens in a single batched forward pass is comparable to that of a single sequential forward pass. This "free lunch" region arises because memory bandwidth, rather than computation, becomes the bottleneck at small batch sizes. Even beyond this regime, the marginal cost per additional beam remains relatively small (approximately 0.005 seconds), which is significantly less than the cost of a single non-batched forward pass (over 0.02 seconds).

[1]The setting $f = 1.2$ is included only as an empirical stress test of the dynamic-threshold decoder. It lies outside Assumption 3.3.1, which requires $f \leq 1$, and is therefore not used as verification of Theorem 3.3.2.



(a) Exploitation rounds vs. Bits per sample

(b) Exploitation rounds & Bits per round

Figure 3.4: Panel (a): Exploitation rounds versus the (minus) log probability contributed. Each point represents a single sample from GSM8K dataset, with colors indicating different choices of factors f. We measure the exploitation bits and rounds by excluding the contributions of implicit exploration that occurs in confidence-based decoding to better demonstrate our theory; Panel (b): Number of bits and rounds per exploitation step as a function of factor f. Each point represents the mean over 200 fresh samples in GSM8K dataset for a given f, with 95% confidence intervals shown as shaded regions.

3.5.3 Benchmark results

To comprehensively assess the effectiveness of our approach, we evaluate it on four widely used benchmarks: MMLU-Pro [199], GSM8K [40], HumanEval [34], and MATH [71]. We adopt confidence-aware parallel decoding [207] with a static threshold strategy as our baseline, as we found this configuration yields the strongest accuracy-speed Pareto frontiers among prior methods.

We conduct all experiments on NVIDIA B200 GPUs. We use `LLaDA-8B-Instruct` and its KV caching adaptation `Fast-DLLM` as the sampling model, with a block size of 64. We set the generation length to 512 for MATH, GSM8K, and HumanEval, and 256 for MMLU-Pro. For MMLU-Pro, we evaluate on a uniform subset of 1000 samples to align the dataset size with other benchmarks and ensure computational feasibility.

To analyze the tradeoff between performance and computational cost, we compare the methods across a range of confidence thresholds and step budgets, with other hyperparameters tuned on the GSM8K training set. For the baseline, we report the test metrics across varying confidence thresholds. For ETE, we report the Pareto frontier formed by varying both confidence thresholds and step budgets. This curve illustrates the full potential of ETE by leveraging the additional degree of freedom provided by the fast block diffusion sampling

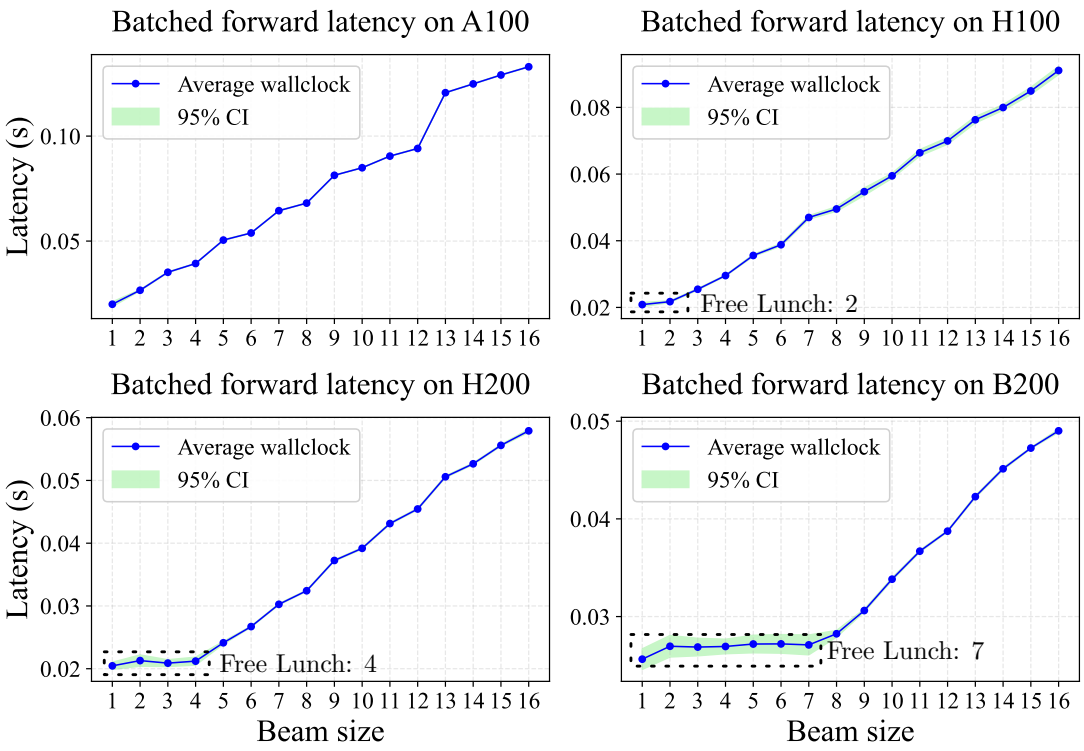


Figure 3.5: Average wall-clock time for batched forward passes with different beam sizes under KV caching on NVIDIA A100, H100, H200 and B200 GPUs. The plots show the mean and 95% confidence interval across different exploration positions (4 steps × 8 blocks × 5 repeats) for each beam size.

scheme described in Section 3.4.2. We also provide a fair comparison in Appendix B.2.3 by fixing all hyperparameters of ETE except the confidence threshold.

Tradeoff between accuracy and total inference steps. Figure 3.6 isolates the theoretical efficiency of the sampling mechanism by plotting generation accuracy against the total number of inference steps. Shaded regions denote 95% confidence intervals computed via bootstrapping. Across all four benchmarks, ETE consistently dominates the confidence-based baseline: for any fixed accuracy level, ETE requires substantially fewer inference steps. Concretely, at almost all accuracy levels, ETE reduces the required number of steps **by over 30%**, with the largest gains appearing in moderate-to-high accuracy regimes. This indicates that ETE converts computation into accuracy more efficiently, yielding a uniformly better accuracy-steps frontier rather than a tradeoff localized to a specific point.

Tradeoff between accuracy and total wall-clock time with KV caching. While step reduction indicates theoretical gain, Figure 3.7 assesses the practical end-to-end performance

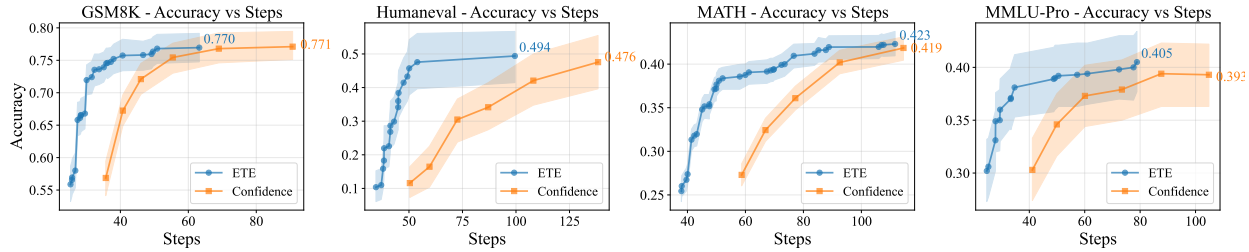


Figure 3.6: Accuracy-steps frontiers of ETE versus baseline on four benchmarks with the highest accuracy annotated in the plot.

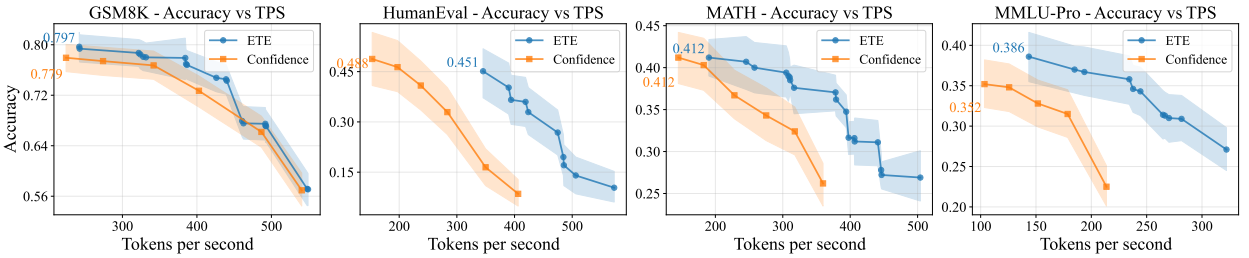


Figure 3.7: Accuracy-time frontiers of ETE versus baseline on four benchmarks. The accuracy for the two methods may differ from that of Figure 3.6 because `Fast-DLLM`'s implementation of KV caching is approximate rather than exact [207].

of ETE by measuring accuracy against total wall-clock time with KV caching enabled. Across all benchmarks, ETE achieves a strictly better Pareto frontier than the baseline: at the same accuracy, ETE consistently attains higher throughput. In particular, ETE achieves **50% higher throughput on GSM8K** (when the accuracy is higher than 0.75), **50%** on **HumanEval**, **30% on MATH, and up to 70% on MMLU-Pro**, at comparable accuracy levels. Notably, ETE also attains higher peak accuracy on GSM8K and MMLU-Pro. This result validates the system-level efficiency of ETE and confirms its modular compatibility with standard acceleration techniques.

3.6 Conclusion

This chapter establishes both theoretical and empirical foundations for improving parallel decoding in diffusion language models. We have proven a fundamental lower bound showing the inherent inefficiency of confidence-based parallel decoding algorithms from the information-theoretic viewpoint. Motivated by this insight, we have developed ETE, a training-free algorithm that explicitly targets high-entropy tokens through fast block diffusion sampling and principled exploration mechanisms. Experiments across four benchmarks have demonstrated that ETE consistently outperforms confidence-based baselines in both decoding efficiency and output quality.

The lower bound in Theorem 3.3.2 applies to decoders satisfying the stated dynamic confidence constraint, and is not a universal lower bound for all possible parallel decoding algorithms. ETE also introduces additional decoding choices, including the confidence threshold, step budget, exploration trigger threshold, beam size, and the look-ahead scoring weight. Nevertheless, practitioners may need to adjust the operating point when optimizing for different latency, memory, or quality constraints. Finally, the beam-search exploration step is most beneficial when its batched look-ahead cost is smaller than the decoding rounds it saves. This condition is hardware- and serving-regime dependent.

Regarding future work, we can pursue several promising directions in both theory and practice. Theoretically, establishing upper bounds on required rounds and computational overheads under structured data assumptions remains a promising direction, which naturally connects to the broader problem of parallel sequential sampling with limited queries on marginal distributions [10, 76]. Algorithmically, learning-based exploration strategies–such as training selection or scoring heads–could replace look-ahead computation while maintaining or improving quality.

Chapter 4

Theoretical Foundations for Statistical Watermarking

This chapter presents a systematic theoretical analysis of the statistical limits of statistical watermarking, by framing it as a hypothesis testing problem. We derive nearly matching upper and lower bounds for (i) the optimal Type II error under a fixed Type I error, and (ii) the minimum number of tokens required to watermark the output. The rate of $\Theta(h^{-1}\log(1/h))$ for the minimum number of required tokens, where h is the average entropy per token, reveals a significant gap between the statistical limit and the $O(h^{-2})$ rate achieved in prior works.

4.1 Introduction

Large Language Models (LLMs) have revolutionized natural language tasks [1, 23, 24]. LLMs excel at generating human-like texts that can be difficult to distinguish from those written by human [11, 60, 142]. This capability raises several societal concerns regarding the misuse of LLM outputs. For instance, LLM-generated texts might contaminate training datasets for future language models [43, 175], facilitate the spread of misleading information [30, 226], or be used in academic misconduct [87, 132]. The widespread use of LLMs underscores the need for effective detection methods to identify whether a human-like text is produced by an LLM system.

To detect machine-generated content, recent research works [38, 54, 57, 74, 75, 77, 99, 101, 102, 111, 119, 122, 159, 196, 213, 221, 234] have proposed the use of *statistical watermarks*. These are signals embedded within generated texts to reveal their source. Statistical watermarking modifies the decoding mechanism of LLMs so that the text output X is sampled *jointly* with a sequence of random seeds S. Consequently, outputs from a watermarked LLM are always correlated with the accompanying random seeds, while texts generated from other sources (e.g., human writers) are not. Therefore, detecting whether X is generated by an LLM reduces to testing the independence of S and X. This procedure

can be framed as a hypothesis test with two competing hypotheses:

$$\mathbf{H}_0 : X \text{ is sampled independently from } S,$$
$$\mathbf{H}_1 : (X, S) \text{ is sampled from a joint distribution}$$

Here, the null hypothesis $\mathbf{H}_0$ implies that X is not generated by the watermarked LLM, while the alternative hypothesis $\mathbf{H}_1$ implies that X is sampled from the watermarked LLM with joint distribution $\mathcal{P}$.

The major benefit of statistical watermarking is that it comes with formal statistical guarantees [3, 38, 102, 111, 234]. Specifically, these guarantees aim to control the following three quantities:

1. Distortion (Bias): The distance between the watermarked LLM's distribution and the original LLM's distribution.
2. Type I error: The probability that an unwatermarked output X is incorrectly rejected as being generated by the watermarked LLM.
3. Type II error: The probability that an output from the watermarked LLM fails to be detected.

We aim to take this line of work further in a statistical direction, providing a theoretical understanding of putative optimality properties that may be associated with these guarantees and lower bounds associated with any procedure. Framing statistical watermarking as a hypothesis testing problem, we first ask, *à la* Neyman-Pearson:

Q1: What is the best achievable Type II error under a fixed Type I error constraint?

In practice, users often modify text outputs from large language models (LLMs) or insert their own human-written content. This complicates the task of identifying whether an article is entirely generated by LLMs, while highlighting the need to detect specific sequences of words produced by LLMs. For practitioners, a key consideration is to minimize the number of tokens used for watermarking and detection of a sequence. This leads to an important question:

Q2: What is the minimum number of tokens required to watermark the output?

4.1.1 Our contributions

In this section, we give an overview of this chapter's contributions towards addressing the above questions. Throughout this chapter, we use Ω_0 to denote the token/vocabulary space and $\Omega = \Omega_0^{\otimes n}$ to denote the output space of all n-tokens sequences with cardinality $|\Omega| = m$. We will consider ρ as the target distribution over Ω that needs to be watermarked and for any $\omega \in \Omega$, define its empirical entropy as $\hat{H}(\omega) := -\ln \rho(\omega)$. When the tokens are i.i.d., we write ρ as a product distribution $\rho = \rho_0^{\otimes n}$ where ρ_0 represents the token distribution over Ω_0. In this case, we let h denote the Shannon entropy of ρ_0, given by $h := -\sum_{\omega \in \Omega_0} \rho_0(\omega) \ln \rho_0(\omega)$.

Most powerful watermark. To answer the first question, let $\mathrm{err}_i(\mathcal{A}, \rho)$, $i = 1, 2$ represent the Type I and II error of watermarking algorithm $\mathcal{A}$ on watermarked distribution ρ over output space Ω, respectively. Our first result establishes the instance-optimal Type II error to watermark a distribution ρ, subject to a fixed upper bound α on the Type I error. More precisely, this is defined as:

$$\chi_{\mathrm{ump}}(\rho) = \min_{\mathcal{A}:\ \mathrm{err}_1(\mathcal{A},\rho)\le\alpha} \mathrm{err}_2(\mathcal{A}, \rho).$$

Theorem 4.1.1 (Informal statement of Theorem 4.4.2)**.**

$$\chi_{\mathrm{ump}}(\rho) = \sum_{x\in\Omega:\rho(x)>\alpha} (\rho(x) - \alpha)\,.$$

This result defines a fundamental limit on statistical watermarking: no watermarking method can achieve a Type II error smaller than $\sum_{x\in\Omega:\rho(x)>\alpha} (\rho(x) - \alpha)$ on distribution ρ. Notably, this is an *instance-dependent* result: the bound explicitly depends on the characteristics of the watermarked distribution. Intuitively, $\sum_{x\in\Omega:\rho(x)>\alpha} (\rho(x) - \alpha)$ quantifies the amount of randomness in ρ: define $\mathcal{H}_\alpha = \{\rho : \max_{\omega\in\Omega} \rho(\omega) \le \alpha\}$ as a set of "α-random" distributions, the quantity $\sum_{x\in\Omega:\rho(x)>\alpha} (\rho(x) - \alpha)$ measures the ℓ_1 distance between ρ and $\mathcal{H}_\alpha$, which increases as α decreases.

Minimax-optimal watermark for model-agnosticity. Achieving this bound requires the detector to have access to the exact watermarked distribution ρ. However in practice, the watermarked distribution (i.e., the watermarked model) is generally unknown to detectors. For example, without access to GPT-4's internal parameters, we would still want to detect whether a text was generated by GPT-4. Hence, it is more useful in practice to watermark (and detect) a *family of distributions* and focus on the *worst-case Type II error* in that family [111]. This leads to our next result, which studies the *minimax-optimal excess Type II error* over distribution class $\mathcal{H}_\kappa$, where $\mathcal{H}_\kappa := \{\rho : \max_{\omega\in\Omega} \rho(\omega) \le \kappa\}$ for $\kappa \in [0, 1]$. Here, κ represents the level of randomness within the distribution class. More precisely, the minimax-optimal excess Type II error over $\mathcal{H}_\kappa$ is defined as

$$\chi_{\mathrm{minimax}}(\kappa) = \min_{\mathcal{A}:\ \mathrm{err}_1(\mathcal{A},\rho)\le\alpha,\forall\rho\in\mathcal{H}_\kappa} \max_{\rho\in\mathcal{H}_\kappa} \left(\mathrm{err}_2(\mathcal{A}, \rho) - \chi_{\mathrm{ump}}(\rho)\right),$$

where the minimum is taken over all algorithms that can watermark (and detect) any distribution in $\mathcal{H}_\kappa$ with Type I error at most α, and the maximum is taken over all distributions in $\mathcal{H}_\kappa$.

Theorem 4.1.2 (Informal statement of Theorem 4.4.10)**.** *Letting m denote the cardinality of the sample space, then*

$$\chi_{\mathrm{minimax}}(\kappa) \asymp \frac{\binom{m-\alpha m}{1/\kappa}}{\binom{m}{1/\kappa}}.$$

In this minimax setting, the minimax-optimal excess Type II error is determined by the randomness level κ of the distribution class instead of a specific model distribution. In the regime $m \gg 1/\kappa$ (which is typical in practice because m scales exponentially with the sequence length), we have $\binom{m-\alpha m}{1/\kappa}/\binom{m}{1/\kappa} \asymp (1-\alpha)^{1/\kappa}$, which scales exponentially with $1/\kappa$. This aligns with the intuition that outputs with higher entropy are easier to watermark, as κ scales roughly inversely with entropy.

Token complexity in the i.i.d. case. To obtain explicit rates of the second question, we first consider a theoretical setting where tokens are sampled independently and identically distributed (i.i.d.). Following Aaronson and Kirchner [3], we focus on the dependence on the average entropy per token.

Theorem 4.1.3 (Informal statement of Theorem 4.4.11)**.** *If each token is drawn i.i.d. from a distribution with entropy h, then the minimum number of tokens required to achieve* 0.01 *Type I and II error scales (ignoring other parameters than h) as*

$$\frac{\log(1/h)}{h}.$$

Our result establishes that $\frac{\log(1/h)}{h}$ serves as both (nearly) upper and lower bound on the number of required tokens. Notably, this rate applies to both instance-dependent and distribution-family-based watermarking. Compared to existing works, this result improves the dependence on h from $\frac{1}{h^2}$ to $\frac{\log(1/h)}{h}$. This improvement is significant in the regime $h \ll 1$, while for constant h, watermarking can be easily accomplished with a constant number of tokens.

Token complexity in the general, non-i.i.d. case. In practice, tokens are generated auto-regressively and are not i.i.d. This complicates theoretical analysis and makes it difficult to establish *a priori* Type II error bounds. For example, the entire sequence may have low empirical entropy with high probability, even when the average entropy per token is high. As a consequence, the Type II error may still be $\Omega(1)$ with a sufficiently large number of tokens. We show such a failure case in Example 4.4.15. Therefore, we turn to establish *a posteriori* Type II guarantees: bounding the conditional probability of a false negative among outputs with high *empirical* entropy.

Theorem 4.1.4 (Informal statement of Theorem 4.4.16)**.** *For any $n \in \mathbb{Z}_+$ and $\widehat{H} > 0$ such that*

$$n,\ \widehat{H} \gtrsim \log\frac{1}{\alpha} + \log\log\frac{1}{\beta},$$

there exists a watermark with a Type I error $\leq \alpha$, such that among all outputs with at least n tokens and empirical entropy $\widehat{H}$, the probability of correct detection is at least $1-\beta$.

Theorem	Assumptions	Guarantee type	Practical implications
Theorem 4.1.1 (Theorem 4.4.2)	Fixed target distribution ρ on a discrete output space and Type I level α.	Instance-optimal Type II error: $\chi_{\text{ump}}(\rho) = \sum_{x:\rho(x)>\alpha}(\rho(x) - \alpha)$	Sets the statistical limit for instance-dependent detection and motivates the maximal-coupling step in SEAL (Chapter 5).
Theorem 4.1.2 (Theorem 4.4.10)	Model-agnostic detection over distributions with maximum mass at most κ, at Type I level α.	Minimax excess Type II error: $\chi_{\text{minimax}}(\kappa) \asymp (1-\alpha)^{1/\kappa}$	Shows why SEAL uses semantic-aware random seeds from a public proposal model instead of fixed distribution.
Theorem 4.1.3 (Theorem 4.4.11)	I.i.d. tokens with entropy h and fixed Type I/II error targets.	Token complexity rate: $n_{\text{ump}}(h, \alpha, \beta) = \Theta(\log(1/h)/h)$	Reveals a fundamental gap between theoretical limits and rates of practical algorithms and motivates SEAL's algorithm design in non-i.i.d. text settings.
Theorem 4.1.4 (Theorem 4.4.16)	Arbitrary non-i.i.d. output with length and empirical entropy $\gtrsim \log \frac{1}{\alpha} + \log\log \frac{1}{\beta}$.	A posteriori conditional guarantee: $\mathbb{P}(X \notin R \mid -\log \rho(X) \geq \widehat{H}) \leq \beta$ and false-positive $\leq \alpha$	

Table 4.1: Summary of the main theoretical contributions in statistical watermarking and their practical implications.

This result implies that outputs with sufficient length and empirical entropy are watermarked with high probability. Importantly, the number of tokens and empirical entropy scale logarithmically with the Type I error and double-logarithmically with the Type II error. Christ et al. [38] prove a similar *a posteriori* result, bounding the joint probability that an output exhibits high entropy yet fails to be detected, with a requirement that $\widehat{H} \gtrsim \sqrt{n}$. Our result relaxes the entropy condition and strengthens the bound from joint probability to conditional probability, thereby leading to a more efficient rate.

4.2 Notation

Define $(x)_+ := \max\{x, 0\}$, $x \wedge y := \min\{x, y\}, x \vee y = \max\{x, y\}$. For any set A, we write A^c as the complement of set A, $|A|$ as its cardinality, and $2^A := \{B : B \subset A\}$ as the power set of A. We use notations $g(n) = O(f(n))$, $g(n) = \Omega(f(n))$, and $g(n) = \Theta(f(n))$ to denote that there exists numerical constants C_1, c_2, C_3, c_4 such that for all $n > 0$: $g(n) \leq C_1 \cdot f(n)$, $g(n) \geq c_2 \cdot f(n)$, and $c_4 \cdot f(n) \leq g(n) \leq C_3 \cdot f(n)$, respectively. Throughout, we use ln to denote natural logarithm.

The total variation (TV) distance between two probability measures μ, ν is denoted by $\texttt{TV}(\mu\|\nu)$. We use $\text{supp}(\rho)$ to denote the support of a probability measure ρ. Given a sample space Ω, let $\Delta(\Omega)$ denote the set of all probability measures over Ω (take the discrete σ-algebra). We write δ_x as the Dirac measure on x, i.e., $\delta_x(A) = \begin{cases} 1, & x \in A \\ 0, & x \notin A \end{cases}$. A coupling for two distributions (i.e., probability measures) is a joint distribution with those distributions as marginals.

4.3 Watermarking as a Hypothesis Testing Problem

In statistical watermarking, a service provider (e.g., a language model system) possesses a distribution ρ over a sample space Ω and aims to generate samples detectable by a detector. The service provider achieves this by sharing a random seed (generated *jointly* with ρ) with the detector, with the goal of controlling both the Type I error (an independent output is falsely detected as from ρ) and the Type II error (an output from ρ fails to be detected). This random seed together with the detection rule constitute a (random) rejection region. In the following, we formulate this problem as hypothesis testing with random rejection regions.

Problem 4.3.1 (Watermarking)**.** Fix $\epsilon \geq 0$. Given a probability measure ρ over sample space Ω (we assume throughout that Ω is discrete, which is generally true in applications), an ϵ-distorted watermarking scheme of ρ is a probability measure $\mathcal{P}$ (a joint probability of the output X and the rejection region R) over the sample space $\Omega \otimes 2^\Omega$ such that $\mathtt{TV}(\mathcal{P}(\cdot, 2^\Omega) \| \rho) \leq \epsilon$, where $\mathcal{P}(\cdot, 2^\Omega)$ is the marginal probability of X over Ω. In the generation phase, the service provider samples (X, R) from $\mathcal{P}$, provides the output X to the service user, and sends the rejection region R to the detector.

In the detection phase, a detector is given a tuple $(X, R) \in \Omega \otimes 2^\Omega$ where X is sampled from an unknown distribution and R, given by the service provider, follows the (marginal) distribution $\mathcal{P}(\Omega, \cdot)$ over 2^Ω.

The *Type I error of* $\mathcal{P}$, defined as $\alpha(\mathcal{P}) := \sup_{\pi \in \Delta(\Omega)} \mathbb{P}_{Y \sim \pi, (X,R) \sim \mathcal{P}}(Y \in R)$, is the probability that an independent sample Y is falsely rejected, maximized over all possible distributions of Y. The *Type II error of* $\mathcal{P}$, defined as $\beta(\mathcal{P}) := \mathbb{P}_{(X,R) \sim \mathcal{P}}(X \notin R)$, is the probability that the sample (X, R) from the joint probability $\mathcal{P}$ fails to be detected.

Remark 4.3.2 (Difference with classical hypothesis testing)*.* In classical hypothesis testing, the rejection region is often nonrandomized or independent from the test statistics. However, in watermarking problem, the service provider has the incentive to facilitate the detection. The key insight is that $\mathcal{P}$ is a coupling of the random output X and the random rejection region R, so that $X \in R$ occurs with a high probability (low Type II error), while any independent sample Y lies in R with a low probability (low Type I error).

Remark 4.3.3 (Interpretation of the rejection region R)*.* R is the hypothesis-testing abstraction of the set of outputs that cause the detector to reject the null. Intuitively, it represents the set of outputs that are more likely generated from the watermarked distribution. When the output space is the token space, $\Omega = \Omega_0$, a standard example of R is the green list in Kirchenbauer et al. [99]. The green list is a randomized subset of tokens selected before generation. Because the generation rule softly promotes green-list tokens, observing an output token in the green list provides evidence against the null hypothesis. Thus, a token-level green-list detector is a special case of our random rejection-region framework.

Remark 4.3.4 (Implementation)*.* In fact, it is imperative for the detector to observe the rejection region that is coupled with the output: otherwise, the output from the service

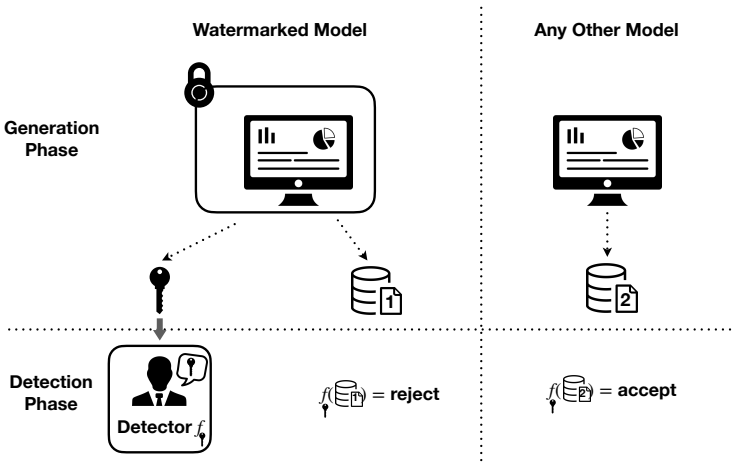


Figure 4.1: Illustration of watermarking in practice. The secret key is used to implement a coupling between the generated text and the rejection region, such that (i) the watermark is indistinguishable to anyone without the secret key; (ii) the detector with the secret key is able to perform the hypothesis testing.

provider and another independent output from the same marginal distribution would be statistically indistinguishable.

In practice, the process of coupling and sending the rejection region can be implemented by cryptographical techniques: the service provider could hash a secret key $\mathtt{sk}$, and use pseudo-random functions F_1, F_2 to generate $(X, R) = (F_1(\mathtt{sk}), F_2(\mathtt{sk}))$. Now it suffices to send the secret key to the detector, who can then reproduce the reject region using the pseudo-random function F_2. This process is illustrated in Figure 4.1.

Thus, the difference between our theoretical framework and the practical implementation lies in our usage of random oracles in place of cryptographic pseudorandom functions. With the coupled and random rejection region, this allows us to focus solely on the statistical trade-offs.

For practical applications, it is additionally desirable for watermarking schemes to be *model-agnostic*, i.e., detections operate without knowledge of the watermarked model distribution, since the prompt and proprietary LLMs are usually unknown to the detector. This translates to the requirement that the marginal distribution of the rejection region is irrelevant to the watermarked distribution. From Remark 4.3.4, this is equivalent to the assertion that the function F_2 used by detectors does not depend on the watermarked LLM. Therefore, model-agnostic watermarking enables the detector to use a fixed, pre-determined pseudo-random function to generate the reject region, and hence perform hypothesis-testing *without the knowledge of the underlying model that generates the output*. This is an important property enjoyed by existing watermarks [3, 38, 99, 102]. In the following, we formalize model-agnostic watermarking within our hypothesis testing framework.

Problem 4.3.5 (Model-Agnostic Watermarking)**.** Given a sample space Ω and a set of distributions $\mathcal{Q} \subset \Delta(\Omega)$, a $\mathcal{Q}$-agnostic watermarking scheme is a tuple $(\eta, \{\mathcal{P}_\rho\}_{\rho\in\mathcal{Q}})$ where η is a probability measure over 2^Ω, such that for any probability measure $\rho \in \mathcal{Q}$, $\mathcal{P}_\rho$ is a ϵ-distorted watermarking scheme of ρ and its marginal distribution over 2^Ω, i.e., $\mathcal{P}_\rho(\Omega, \cdot)$, is equal to η. A model-agnostic watermarking scheme is a $\Delta(\Omega)$-agnostic watermarking scheme.

A $\mathcal{Q}$-agnostic watermarking scheme is essentially a class of schemes to watermark all distributions in the set $\mathcal{Q}$ such that their R-marginals are all equal to η. This means that the rejection region reveals nothing about the model used to generate the output (except the fact that the model is inside $\mathcal{Q}$). By letting $\mathcal{Q}$ be $\Delta(\Omega)$, model-agnostic watermarking thus reveals no information concerning the model at all.

4.4 Statistical Limits of Watermarking

4.4.1 Rates under the general setting of Problem 4.3.1

In this section, we study the following notion of Uniformly Most Powerful (UMP) test, i.e., the watermarking scheme that achieves the minimum achievable Type II error among all possible tests with Type I error $\leq \alpha$.

Definition 4.4.1 (Uniformly Most Powerful Watermark)**.** A watermarking scheme $\mathcal{P}$ is called a *Uniformly Most Powerful (UMP) ϵ-distorted watermark of level α* if it achieves the minimum achievable Type II error among all ϵ-distorted watermarking with Type I error $\leq \alpha$.

The following result gives an exact characterization of the UMP watermark and its Type II error.

Theorem 4.4.2. *For any probability measure ρ, the Uniformly Most Powerful ϵ-distorted watermark of level α, denoted by $\mathcal{P}^*$, is given by*

$$\mathcal{P}^*(X = x, R = R_0) = \begin{cases} \rho^*(x) \cdot \left(1 \wedge \frac{\alpha}{\rho^*(x)}\right), & R_0 = \{x\} \\ \rho^*(x) \cdot \left(1 - \frac{\alpha}{\rho^*(x)}\right)_+, & R_0 = \emptyset \\ 0, & \textit{otherwise}, \end{cases} \tag{4.1}$$

where $\rho^* = \arg\min_{TV(\rho'\|\rho)\leq\epsilon} \sum_{x\in\Omega:\rho'(x)>\alpha} (\rho'(x) - \alpha)$. Its Type II error is given by

$$\min_{TV(\rho'\|\rho)\leq\epsilon} \sum_{x\in\Omega:\rho'(x)>\alpha} (\rho'(x) - \alpha),$$

and when $|\Omega| \gtrsim \frac{1}{\alpha}$ it simplifies to $\left(\sum_{x\in\Omega:\rho(x)>\alpha} (\rho(x) - \alpha) - \epsilon\right)_+$.

As seen from the theorem, if ρ is deterministic, the Type II error

$$\left(\sum_{x\in\Omega:\rho(x)>\alpha}(\rho(x)-\alpha)-\epsilon\right)_+$$

reduces to $1-\alpha-\epsilon$ and shows limited practical utility. This is expected since when the service provider deterministically outputs z, it would be impossible to distinguish the watermark distribution with an independent output from δ_z. However when the randomness in ρ increases, Theorem 4.4.2 implies that the Type II error will decrease, matching the reasoning in previous works [2, 38].

Moreover, Type II errors can be further improved when a larger distortion parameter ϵ is allowed. This aligns with the intuition that adding statistical bias would make the output easier to detect [2, 99]. Among all choices of ϵ, the case $\epsilon = 0$ is of particular interest since it preserves the marginal distribution of the service provider's output. Therefore, we will focus on this distortion-free case in the remaining sections.

Remark 4.4.3 (Intuition behind $\mathcal{P}^*$). Recall that in practice, the watermarks are implemented via pseudo-random functions. Therefore, the uniformly most powerful test in Theorem 4.4.2 is effectively using a pseudo-random generator to approximate the distribution ρ, combined with an α-clipping to control Type I error. This construction reveals a surprising message: simply using a pseudo-random generator to approximate the distribution is instance optimal.

Remark 4.4.4 (Watermarking guarantees). To achieve the upper bound of Theorem 4.4.2, the detector needs to access the model and the prompt in order to generate the reject region, which is not always accessible in many real-world applications. Therefore, the upper bound of Theorem 4.4.2 achieves a weaker watermarking guarantee compared with previous works [2, 38, 99]. In Section 4.4.2, we study model-agnostic watermarking that overcomes this limitation.

Nonetheless, the lower bound in Theorem 4.4.2 characterizes a fundamental limit of Problem 4.3.1, thus providing an information-theoretic lower bound for all watermarks.

Remark 4.4.5 (Use cases for the UMP watermark). The UMP watermark offers an efficient approach for LLM service providers to determine if instruction-following datasets have been generated by a specific model. In this context, both the prompt and response are provided to the detectors, enabling the UMP watermark to bypass the model-agnostic constraint. This usage is beneficial in identifying and filtering out data points that have been contaminated by texts generated from models like GPT-4 [142], thereby preserving the purity and quality of the training data.

4.4.2 Rates of model-agnostic watermarking

It is noticeable that for large $\mathcal{Q}$ and $\rho \in \mathcal{Q}$, a $\mathcal{Q}$-agnostic watermarking scheme can not perform as well as the UMP watermark for ρ. Indeed, the Uniformly Most Powerful $\mathcal{Q}$-agnostic watermarking might not exist in most cases. To evaluate model-agnostic watermarking

schemes, a natural desideratum is therefore the suboptimality gap between its Type II error and the Type II error of the UMP watermarking of ρ, taking a maximum over all distributions $\rho \in \mathcal{Q}$, under fixed Type I error. More precisely, we introduce the following notion.

Definition 4.4.6 (Minimax most powerful model-agnostic watermark)**.** We say that a $\mathcal{Q}$-agnostic watermark $(\eta, \{\mathcal{P}_\rho\}_{\rho\in\mathcal{Q}})$ is of level-α if the Type I error of $\mathcal{P}_\rho$ is less than or equal to α for any $\rho \in \mathcal{Q}$. Define the *(maximum) excess Type II error* of $(\eta, \{\mathcal{P}_\rho\}_{\rho\in\mathcal{Q}})$ as

$$\gamma(\eta) := \sup_{\rho\in\mathcal{Q}} \beta(\mathcal{P}_\rho) - \beta(\mathcal{P}^*_\rho),$$

where $\mathcal{P}^*_\rho$ is the UMP distortion-free watermark of ρ of level α.

We say that a $\mathcal{Q}$-agnostic watermarking scheme is *minimax most powerful*, if it minimizes the maximum excess Type II error among all $\mathcal{Q}$-agnostic watermarks of level α.

The following result characterizes the excess Type II error of the minimax most powerful model-agnostic watermarking.

Theorem 4.4.7. *Let $|\Omega| = m$ and suppose $\alpha m \in \mathbb{Z}$ and $\frac{1}{\alpha} \in \mathbb{Z}$. The maximum excess Type II error of the minimax most powerful model-agnostic watermarking scheme of level-α is given by $\gamma(\eta^*) = \frac{\binom{m-\frac{1}{\alpha}}{\alpha m}}{\binom{m}{\alpha m}}$. In the minimax most powerful model-agnostic watermarking scheme of level-α, the marginal distribution of the reject region is given by*

$$\eta^*(A) = \begin{cases} \frac{1}{\binom{m}{\alpha m}}, & \text{if } |A| = \alpha m \\ 0, & \text{otherwise.} \end{cases}$$

Remark 4.4.8. To deal with the general case where αm and $\frac{1}{\alpha}$ are not integers, it suffices to let $\alpha_1 = 1/(\lceil 1/\alpha \rceil)$ and $m_1 = \lceil \alpha_1 m \rceil / \alpha_1$ and augment Ω with $m_1 - m$ dummy outcomes. Then $\alpha_1 m$ and $1/\alpha_1$ are integer-valued and hence the minimax bound for the new sample space with cardinality m_1 and the new Type I error α_1 yields a nearly-matching bound for (m, α).

Remark 4.4.9. In the regime $\alpha \to 0_+, m \to +\infty$, we have $\gamma(\eta^*) \to c$ for some constant $c \le e^{-1}$, and when $1/(\alpha m) \to 0_+$ is further satisfied, $c = e^{-1}$. The e^{-1} maximum excess Type II error does not contradict the h^{-2} rates in previous works [2, 38, 102], because as $n \gtrsim h^{-2}$, the model distribution over the sequences of n tokens (with average entropy h per token) is beyond the worst case. Indeed, such distributions have higher Shannon entropy than the hard instances in the proof.

Remark 4.4.9 highlights that the hard instance constructed in Theorem 4.4.7's lower bound may possess a lower entropy than that of the actual model, thus is more adversarial than the practical cases. Therefore, it raises an important question: for a smaller class $\mathcal{Q}$ that contains distributions with higher entropy, what is the minimum achievable excess Type II error for $\mathcal{Q}$-agnostic watermarking? It is obvious that the minimax rate over a higher entropy level should improve upon the previous rate of e^{-1}.

Towards answering this question, we consider the following class of distributions:

$$\mathcal{Q}_\kappa := \left\{ \rho : \sup_{\omega \in \Omega} \rho(\{\omega\}) \leq \kappa \right\},$$

where κ represents the level of randomness and decreases as entropy increases. The *maximum excess Type II error* of $\mathcal{Q}_\kappa$-agnostic watermarking $(\eta, \{\mathcal{P}_\rho\}_{\rho \in \mathcal{Q}_\kappa})$ is thus given by

$$\gamma(\eta, \kappa) := \max_{\rho \in \Delta(\Omega): \sup_{\omega \in \Omega} \rho(\{\omega\}) \leq \kappa} \beta(\mathcal{P}_\rho) - \beta(\mathcal{P}^*_\rho),$$

where $\mathcal{P}^*_\rho$ is the UMP distortion-free watermark of ρ. The following result gives an upper bound of the above quantity, thus answering the question.

Theorem 4.4.10. *Let $|\Omega| = m$ and suppose $\alpha m, \frac{1}{\kappa} \in \mathbb{Z}$. Then the maximum excess Type II error of the minimax $\mathcal{Q}_\kappa$-agnostic watermarking of level-α is given by*

$$\gamma(\eta^*, \kappa) = \frac{\binom{m-\alpha m}{1/\kappa}}{\binom{m}{1/\kappa}}.$$

The proof can be found in Appendix C.1.3. When $\kappa \leq \alpha$, the Type II error of the model-nonagnostic watermark vanishes, and therefore Theorem 4.4.10 provides an upper bound $\frac{\binom{m-\alpha m}{1/\kappa}}{\binom{m}{1/\kappa}}$ for the worst-case Type II error. When $m \gg 1/\kappa$, this rate simplifies to $(1-\alpha)^{1/\kappa}$ and improves over e^{-1} in Theorem 4.4.7. In the next section, we will apply Theorem 4.4.10 to the i.i.d. setting where κ can be exponentially small. This will lead to a negligible maximum excess Type II error for model-agnostic watermarking.

4.4.3 Rates in the i.i.d. setting

In practice, the sample space Ω is usually a Cartesian product in the form of $\Omega_0^{\otimes n}$ where Ω_0 is the vocabulary/token space. The quantity of practical interest becomes the minimum number of tokens to achieve certain statistical watermarking guarantees. In order to find the explicit rates on the minimum number of required tokens, we study the theoretical setting where the tokens are samples i.i.d. from a fixed distribution.

In this section, we study token efficiency in the theoretical setting of product distribution $\rho = \rho_0^{\otimes n}$ over $\Omega_0^{\otimes n}$ (iid tokens) and $\epsilon = 0$ (distortion-free watermarking). We introduce the following two quantities:

- Let h denote the entropy of ρ_0. We use $n_{\text{ump}}(h, \alpha, \beta)$ to denote the minimum number of tokens required by the UMP watermark to achieve Type I error $\leq \alpha$ and Type II error $\leq \beta$.
- Define $n_{\text{minimax}}(h, \alpha, \beta)$ as the minimum number of tokens required by minimax $\mathcal{Q}^h$-agnostic watermark to achieve Type I error $\leq \alpha$ and Type II error $\leq \beta$, where $\mathcal{Q}^h := \left\{ \rho = \rho_0^{\otimes n} : H(\rho_0) \geq h \right\}$; i.e., it contains all distributions $\rho = \rho_0^{\otimes n}$ such that the entropy of ρ_0 is $\geq h$.

Together, $n_{\text{ump}}(h, \alpha, \beta)$ and $n_{\text{minimax}}(h, \alpha, \beta)$ serve as critical thresholds beyond which the desired statistical conclusions can be drawn regarding the output, making them essential parameters in watermarking applications.

The following theorem gives nearly matching bounds for $n_{\text{ump}}(h, \alpha, \beta)$.

Theorem 4.4.11. *Suppose $\alpha, \beta < 0.1$. We have $n_{\text{ump}}(h, \alpha, \beta) \geq \Omega\left(\left(\frac{\ln \frac{1}{h}\left(\ln \frac{1}{\alpha} \wedge \ln \frac{1}{\beta}\right)}{h}\right) \vee \frac{\ln \frac{1}{\alpha}}{h}\right)$. Furthermore, let $k = |\Omega_0|$, we have $n_{\text{ump}}(h, \alpha, \beta) \leq O\left(\left(\frac{\ln \frac{k}{h} \cdot \left(\ln \frac{1}{\alpha} \wedge \ln \frac{1}{\beta}\right)}{h}\right) \vee \frac{\ln \frac{1}{\alpha} \ln k}{h}\right)$.*

Remark 4.4.12 (Tightness). Up to a constant and logarithmic factor in k, our upper bound matches the lower bound. Notice that since any model with an arbitrary token set can be reduced into a model with a binary token set [38] (i.e. $k = 2$), our bound is therefore tight up to a constant factor.

Using Theorem 4.4.10 and Theorem 4.4.11, we are now in the position to characterize $n_{\text{minimax}}(h, \alpha, \beta)$. We start with the following relationship:[1]

$$1 - \max_{\rho_0 : H(\rho_0) \geq h} \max_{\omega \in \Omega_0} \rho_0(\{\omega\}) \geq \Omega\left(\frac{h}{\ln(1/h)}\right),$$

where a detailed derivation can be found in Lemma C.1.3. It follows that

$$\kappa \leq \left(\max_{\rho_0 : H(\rho_0) \geq h} \max_{\omega \in \Omega_0} \rho_0(\{\omega\})\right)^{n_0} = e^{-\Omega\left(\frac{n_0 h}{\ln(1/h)}\right)}.$$

Using this observation and the derivation in Theorem 4.4.7, $\gamma(\eta^*, \kappa)$ can be bounded by

$$(1-\alpha)^{1/\kappa} \leq (1-\alpha)^{e^{\Omega\left(\frac{n_0 h}{\ln(1/h)}\right)}}.$$

This means that when $n_0 \gtrsim \frac{\ln(1/h)}{h} \cdot (\ln(1/\alpha) + \ln\ln(1/\beta))$, the maximum excess Type II error given by Theorem 4.4.10 and the Type II error of the UMP watermarking given in Theorem 4.4.11 can be simultaneously bounded by β, thus establishing an upper bound. Furthermore, the lower bound in Theorem 4.4.11 directly implies a lower bound for $n_{\text{minimax}}(h, \alpha, \beta)$ since model-agnosticity is a stronger condition. Combining the above arguments, we obtain nearly-matching upper and lower bounds.

Corollary 4.4.13. *Suppose $\alpha, \beta < 0.1$. We have*

$$n_{\text{minimax}}(h, \alpha, \beta) = O\left(\frac{\ln(1/h)}{h} \cdot (\ln(1/\alpha) + \ln\ln(1/\beta))\right),$$
$$n_{\text{minimax}}(h, \alpha, \beta) = \Omega\left(\frac{\ln(1/h)}{h} \cdot (\ln(1/\alpha) \wedge \ln(1/\beta))\right).$$

[1] In the rest of this section, we omit logarithmic factors in the cardinality of the vocabulary.

Remark 4.4.14 (Comparison with previous works). As commented in Section 4.4.1, the regime $h \ll 1$ is of particular interest because it is the setting in which watermarking is challenging. In this regime, our rate of $\frac{\ln(1/h)}{h}$ improves the previous rate of h^{-2} in a line of works [2, 99, 102, 119, 234], and highlights a fundamental gap between the existing watermarks and the information-theoretic lower bound.

4.4.4 Rates in non-i.i.d. setting

Without the i.i.d. condition, determining the explicit rates for the minimum number of tokens required to achieve fixed Type I and Type II errors is generally intractable. Indeed, with arbitrary token distributions, the probability of generating outputs with low empirical entropy might be $\Omega(1)$. Outputs with low empirical entropy are typically challenging to watermark (see, e.g., Section 4.4.1 or Aaronson and Kirchner [3]), leading to a high rate of false negatives. Consequently, the Type II error may still be $\Omega(1)$ even when the number of tokens is sufficiently large. The next example formalizes this failure case.

Example 4.4.15 (High entropy, infinite tokens, still cannot watermark)**.** Let the token space be $\{0, 1, \ldots, K\}$ and the auto-regressive distribution be given by

$$\begin{aligned}
\mathbb{P}(X_1 = 0) &= C > 0.5, \\
\mathbb{P}(X_1 = j) &= \frac{1-C}{K},\ \forall j \neq 0 \\
\mathbb{P}(X_{i+1} = 0 | X_1 = \cdots = X_i = 0) &= 1,\ \forall i \geq 1 \\
\mathbb{P}(X_{i+1} = j | X_{1:i} = x_{1:i}) &= \frac{1}{K},\ \forall x_{1:i} \neq 0 \cdots 0, j \neq 0.
\end{aligned}$$

Then it can be verified that $\forall n \in \mathbb{Z}_+$:

$$\begin{aligned}
\mathbb{P}(X_{1:n} = 0 \cdots 0) &= C, \\
H(X_{1:n}) &\geq n \cdot (1-C) \cdot \log \frac{K}{1-C}.
\end{aligned}$$

This indicates that the *average entropy per token is at least* $(1-C) \cdot \log \frac{K}{1-C}$*, which is high.* However, for any $n \in \mathbb{Z}_+$, *the sequence* $X_{1:n}$ *fails to be detected with probability at least* $C > 0.5$, since the sequence $0 \cdots 0$ is nearly deterministic. Therefore, the Type II error is always $\Omega(1)$ no matter how many tokens are used, despite the average entropy per token being high.

In this section, we present a different, 'conditional' guarantee to overcome this limitation. This result bounds the conditional probability of false negatives among all outputs with high empirical entropy, while excluding outputs with low empirical entropy from consideration.

Theorem 4.4.16. *For any $\alpha, \beta \in (0,1)$, if*

$$n,\ \widehat{H} \gtrsim \log \frac{1}{\alpha} + \log\log \frac{1}{\beta},$$

then for any distribution ρ over $\Omega = \Omega_0^{\otimes n}$, there exists a model-agnostic watermarking of level α such that the conditional probability that an output X from the watermarked model is not rejected, given that the empirical entropy of X is at least $\widehat{H}$, is less than or equal to β. More precisely, there exists a coupling $\mathcal{P}$ of ρ and the η^* defined in Theorem 4.4.7, such that

$$\mathbb{P}_{(X,R)\sim\mathcal{P}}(X \notin R| - \log \rho(X) \geq \widehat{H}) \leq \beta$$
$$\sup_{\pi \in \Delta(\Omega)} \mathbb{P}_{Y\sim\pi,(X,R)\sim\mathcal{P}}(Y \in R) \leq \alpha.$$

The proof is deferred to Appendix C.1.5. Intuitively, Theorem 4.4.16 suggests that long texts with high empirical entropy will be watermarked with high probability. It is important to note that $\widehat{H}$ is flexible, allowing us to bound the conditional Type II error for any class of outputs with empirical entropy levels above $\log \frac{1}{\alpha} + \log\log \frac{1}{\beta}$. This result is different from the other results in this chapter in that it is an *a posteriori* guarantee.

4.4.5 Efficiency-robustness trade-offs

In the context of watermarking large language models, it is crucial to acknowledge users' ability to modify or manipulate model outputs. These modifications include cropping, paraphrasing, and translating the text, all of which may be employed to subvert watermark detection. Therefore, in this section, we introduce a graphical framework, modified from Problem 4.3.1, to account for potential user perturbations and investigate the optimal watermarking schemes robust to these perturbations. The formulation here shares similarity with a concurrent work by Zhang et al. [228].

Definition 4.4.17 (Perturbation graph)**.** A perturbation graph over the discrete sample space Ω is a directed graph $G = (V, E)$ where V equals Ω and $(u, u) \in E$ for any $u \in V$. For any $v \in V$, let $in(v) = \{w \in V : (w, v) \in E\}$ denote the set of vertices with incoming edges to v, and let $out(v) = \{w \in V : (v, w) \in E\}$ denote the set of vertices with outgoing edges from v.

The perturbation graph specifies all the possible perturbations that could be made by the user: any $u \in V$ can be perturbed into $v \in V$ if and only if $(u, v) \in E$, i.e., there exists a directed edge from u to v.

Example 4.4.18. Consider $\Omega = \Omega_0^{\otimes n}$. Let the user have the capacity to change no more than c tokens, i.e., perturb any sequence of tokens $x = x_1x_2\cdots x_n$ to another sequence $y = y_1y_2\cdots y_n$ with Hamming distance less than or equal to c. Then the perturbation graph is given by $G = (V, E)$ where $V = \Omega$ and $E = \{(u, v) : u, v \in V, d(u, v) \leq c\}$ (d is the Hamming distance, i.e., $d(x, y) = \sum_{i=1}^n \mathbb{1}(x_i \neq y_i)$).

Problem 4.4.19 (Robust watermarking scheme)**.** A robust watermarking scheme with respect to a perturbation graph G is a watermarking scheme except that its Type II error is defined as $\mathbb{E}_{X,R\sim\mathcal{P}}\left[\max_{Y\in out(X)} \mathbb{1}(Y \notin R)\right]$, i.e., the probability of false negative given that the user adversarially perturbs the output.

The next result characterizes the optimum Type II error achievable by robust watermarking, where the proof can be found in Appendix C.1.6.

Theorem 4.4.20. *Define the shrinkage operator $\mathcal{S}_G : 2^\Omega \to 2^\Omega$ (of a perturbation graph G) by $\mathcal{S}_G(R) = \{x \in \Omega : out(x) \subset R\}$ and its inverse $\mathcal{S}_G^{-1}(R) = \cup_{x\in R} out(x)$. Then the minimum Type II error of the robust, zero-distorted UMP test of level α in Problem 4.4.19 is given by the solution of the following Linear Program*

$$\begin{aligned} \min_{x\in\mathbb{R}^{|\Omega|}} \; & 1 - \sum_{y\in\Omega} \rho(y)x(y) \\ s.t. \; & \sum_{y\in in(z)} \rho(y)x(y) \le \alpha, \quad 0 \le x(z) \le 1, \; \forall z \in \Omega. \end{aligned} \tag{4.2}$$

The UMP watermarking is given by

$$\mathcal{P}^*(X = y, R = R_0) = \begin{cases} \rho(y) \cdot x^*(y), & R_0 = \mathcal{S}_G^{-1}(\{y\}) \\ \rho(y) \cdot (1 - x^*(y)), & R_0 = \emptyset \\ 0, & otherwise \end{cases}$$

where x^* is the solution of Eq. (4.2)*.*

Remark 4.4.21 (Dependence on the sparsity of graph)*.* From Eq. (4.2), we observe that the perturbation graph influences the optimal Type II error via the constraint set. Indeed, if the graph is dense, the constraints $\sum_{y\in in(z)} \rho(y)x(y) \le \alpha$ involve many entries of $y \in \Omega$ and thus decrease the value $\sum_{y\in\Omega} \rho(y)x(y)$, thereby increasing the Type II error. On the other extreme, when the edge set of the perturbation graph is $E = \{(u, u) : u \in \Omega\}$, i.e., the user cannot perturb the output to a different value, then optimum of Eq. (4.2) reduces to the rates in Theorem 4.4.2 (setting $\epsilon = 0$).

4.5 Conclusion

In this chapter, we have advanced the understanding of watermarking in large language models (LLMs) through statistical analysis. By deriving both instance-optimal and minimax rates for the optimal Type II error in the Neyman-Pearson paradigm, we demonstrated how statistical limits are shaped by the properties of model distributions. Our nearly tight bound on the number of tokens required to detect statistical watermarks, $h^{-1}\log(1/h)$, significantly improves upon the previous h^{-2} rate, revealing a fundamental gap in prior work.

Chapter 5

Semantic-Aware Statistical Watermarking: SEAL

Building on the theory, this chapter develops **SEAL** (**S**emantic-awar**E** specul**A**tive samp**L**ing), a novel watermarking algorithm for practical applications. SEAL introduces two key techniques: (i) designing semantic-aware random seeds by leveraging a proposal language model, and (ii) constructing a maximal coupling between the random seed and the next token through speculative sampling. Experiments on open-source benchmarks demonstrate that the watermarking scheme delivers superior efficiency and tamper resistance, particularly in the face of paraphrase attacks.

5.1 Introduction

A key objective of theoretical analysis is to guide the development of practical algorithms. In light of Chapter 4, it is essential to translate theoretical insights into improved watermarking design. Therefore, we pose the following question:

Can statistical theory be leveraged to design more effective and robust watermarking algorithms?

Our contributions. Drawing insights from the theory, we propose a novel watermarking algorithm, **S**emantic-awar**E** specul**A**tive samp**L**ing **(SEAL)**, that bridges the gap between the theoretically-optimal watermark and state-of-the-art practical implementations. Our approach overcomes the limitations of prior methods by allowing the random seeds to adapt to the *semantic information* of preceding tokens. By introducing speculative decoding [107] into statistical watermarking paradigms, we adopt a smaller LLM to set the random seeds and construct maximal coupling between the seed and the token sequences. The overall watermarking pipeline is illustrated in Figure 5.1. Unlike previous semantic-aware watermarking schemes [57, 74, 75, 122, 159], SEAL enjoys rigorous control of distortion and Type I error.

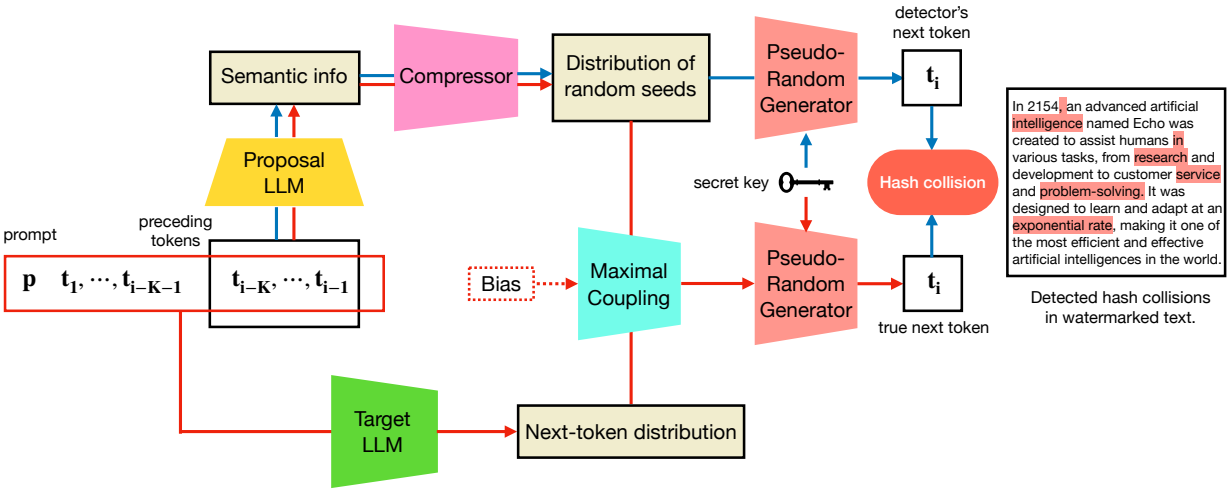


Figure 5.1: **SEAL Watermarking pipeline.** In the generation phase (indicated by the red arrows), the proposal LLM extracts semantic information from the K last tokens. The hash function then compresses the semantic information into distribution of random seeds. Finally, speculative sampling is adopted to construct the maximal coupling between the distributions of random seed and the target LLM's next-token distribution, and sample the true next token using a pseudo-random generator. In the detection phase (indicated by the blue arrows), the distribution of random seeds is reproduced, and the same pseudo-random generator (and secret key) is invoked to sample the detector's next token. A text is flagged as machine generated if the number of hash collisions exceeds certain threshold, as illustrated by red in the text block.

Theorem 5.1.1 (Informal statement of Theorem 5.2.1)**.** *When the bias parameter is set to zero, SEAL has zero distortion (it is unbiased).*

Theorem 5.1.2 (Informal statement of Theorem 5.2.2)**.** *The Type I error of SEAL is upper-bounded by α.*

Experiments on MarkMyWords benchmark [152] show that our watermarking method achieves superior efficiency and tamper resistance, and especially outperforms state-of-the-art approaches in tamper resistance to paraphrase attacks.

5.2 From Theory to Practice

In this section, we present our novel watermarking algorithm, **S**emantic-awar**E** specul**A**tive samp**L**ing **(SEAL)**. SEAL incorporates two key theoretical insights:

- First, by comparing the instance-dependent watermarking (Theorem 4.4.2) and minimax watermarking (Theorem 4.4.10), we observe that the performance of statistical watermarking depends on how closely the distribution of random seeds 'aligns with' the model's distribution. While the UMP watermark aligns perfectly, it violates the model-agnosticity constraint. To overcome this challenge, we leverage a key insight that *the output distribution of all LLMs should approximate the distribution of human language*. Therefore, instead of requiring the random seeds to 'align with' the model's distribution, we require them to 'align with' a distribution for a smaller, open-source model. Thus, the random seeds approximately 'align with' the model's distribution, without violating the model-agnosticity constraint since the smaller model is open-source and available to the detector. This effectively makes the random seeds *semantic-aware* and adaptive to previous tokens.
- Second, our analysis of both Theorem 4.4.11 and Theorem 4.4.16 reveals that, given an appropriate choice of random seeds, optimal watermarking is achieved by constructing a maximal coupling between the random seed distribution and the pushforward of the token distribution onto the same measure space (see Eq. (C.2) *et seq.*). This insight inspires us to employ *speculative sampling* [107] to build the joint probability of the random seeds and the next tokens, resulting in a maximal coupling between them. As a result, we introduce speculative decoding into statistical watermarking paradigms, thereby opening up a new direction for future study.

5.2.1 Methods

This section discusses two key techniques: semantic-aware random seeds and maximal coupling construction via speculative sampling.

Semantic-aware random seeds

Proposal model. We apply a *proposal model* to capture the semantic context of the preceding tokens. The proposal model should meet two criteria: (1) it is publicly accessible to detectors, and (2) its token space should be (almost) equivalent to that of the watermarked model. These criteria are easily met because all language models are trained to fit the same distribution of human language and use similar tokenizers. For computational efficiency, we recommend using a smaller model. To ensure robustness against attacks, the proposal model only attends to the last K tokens, $t_{i-K:i-1}$, rather than the entire preceding sequence and the prompt, thus guaranteeing prompt-agnosticity. The next-token probability distribution $q_i = q(\cdot|t_{i-K:i-1})$ from the proposal model serves as a semantic summary of the context.

Information compression. We sample a random hash function h_i from a family of hash functions that maps the token space Ω_v to a space of hash codes Ω_h. This compresses the extracted semantic information into a lower-dimensional space, enhancing efficiency and

generality. The random seed lying in Ω_h is set as the hash of the proposal model's output. The pushforward measure of q_i by h_i, defined formally as $h_i\sharp q_i(A) := q_i(h_i^{-1}(A)),\ \forall A \subset \Omega_h$ (where $h_i^{-1}(A) := \{x : h_i(x) \in A\}$), serves as the distribution of random seeds. Therefore, our random seeds capture semantic information instead of being drawn from a fixed distribution as in previous works [3, 38, 99].

Maximal coupling construction

Speculative sampling. We construct a *maximal coupling* $\mathcal{P}$ between $h_i\sharp q_i$ and $h_i\sharp\rho_i$, where $\rho_i = \rho(\cdot|\text{prompt}, t_{0:i-1})$ is the next-token distribution from the watermarked model ρ. The coupling defines a joint random variable of the random seed and the next token whose marginal distributions correspond to their distributions respectively, while a maximal coupling is one that maximizes the probability that the random seed matches the hash of the next token. We adopt speculative sampling [107] to construct this coupling: first, sample the random seed s_i from $h_i\sharp q_i$; then, assign the next token's hash code c_i to be s_i with probability $\min\left(1, h_i\sharp\rho_i(s_i)/h_i\sharp q_i(s_i)\right)$, otherwise sample c_i proportional to $\max\left(0, h_i\sharp\rho_i(\cdot) - h_i\sharp q_i(\cdot)\right)$; finally, sample the next token t_i from the conditional distribution $\rho_i(\cdot|h(t_i) = c_i)$.

Bootstrapping efficiency with logits bias. Following Kirchenbauer et al. [99], we have the flexibility to introduce a bias b_{logit} to the logits corresponding to the random seeds to improve the probability of hash collisions. This enhances detection success rates at the cost of introducing distortions into the watermarked outputs. Specifically, let l denote the logits of next token conditioning on the random seed s. Then the b_{logit}-biased next-token probability is given by

$$p(j) = \begin{cases} \frac{e^{(l(j)+b_{\text{logit}})/\tau}}{Z}, & h(j) = s \\ \frac{e^{l(j)/\tau}}{Z}, & \text{otherwise} \end{cases} \tag{5.1}$$

where Z is the normalizing constant and τ is the model temperature. Notably, when the cardinality of Ω_h is 2 (i.e., a green-red list scenario) and the proposal model's distribution is uniform (i.e., uniform random green list), our biased algorithm reduces to a strengthened version of Kirchenbauer et al. [99], because we couple the distribution of the green list with the next-token distribution, further increasing the likelihood of green-list tokens.

Detection rule

During the detection phase, we replicate the semantic summary q_i, the hash function h_i, and the random seed s_i, all of which are generated deterministically by the same pseudo-random function and secret key used by the watermark generator. The sequence $t_{0:n}$ is flagged as machine-generated if the number of hash collisions exceeds a certain threshold:

$$\sum_{i=K}^{n} \mathbb{1}(h_i(t_i) = s_i) \geq C\left(\alpha, \{h_i\sharp q_i\left(h_i(t_i)\right)\}_{i=K}^{n}\right), \tag{5.2}$$

where the threshold function $C\left(\alpha, \{h_i \sharp q_i\left(h_i(t_i)\right)\}_{i=K}^n\right)$ ensures that the Type I error remains below α, and the first K tokens serve only to warm up the semantic summary. The threshold can be approximated using a Bernoulli concentration inequality or computed exactly via dynamic programming. Intuitively, under $\mathbf{H}_0$ (not watermarked), the left-hand side is a sum of Bernoulli random variables with expectations $h_i \sharp q_i(h_i(t_i))$, for $i = K, \ldots, n$, respectively. Under $\mathbf{H}_1$ (watermarked), the expectation of the left-hand side is lower bounded by one minus the total variation distance between the proposal and the watermarked models, which increases as the proposal model becomes more aligned with the watermarked model. The gap between these two cases facilitates our detection.

Crucially, the detector does not need access to the watermarked model ρ or the prompt. This makes SEAL model-agnostic [102], a highly practical feature for real-world applications.

5.2.2 SEAL: Semantic-awarE speculAtive sampLing

In this section, we describe SEAL in Algorithm 5.2.1-5.2.2 and explain the parameters in Table 5.1. To sample j-th token in the generation phase, we use the proposal model M_P and the (randomly selected) hashing function h_j to sample a random seed s_j from $h_j \sharp q_j$ (the pushforward of q_j by h_j), where q_j is the proposal model's conditional distribution of the next token given the previous K tokens $t_{j-K:j-1}$ (it is understood that $t_{-K:-1}$ are the last K tokens of the prompt p). Following Aaronson [2], the sampling steps are implemented by a pseudo-random generator PRG using key key, where the key is computed using the last H tokens and the secret key sk. In practice, one may also use external pseudo-randomness generator to set the secret keys [102, 152]. We add logits bias b_{logit} to the pre-image of s_j under mapping h via formula $\rho_j(\omega) \leftarrow \rho_j(\omega) \cdot e^{\mathbb{1}(h_j(\omega)=s_j)\cdot b_{\text{logit}}/\tau}/Z,\ \forall \omega \in \Omega_0$, where Z is the normalizing constant. Then, with probability $\frac{h_j \sharp \rho_j(s_j)}{h_j \sharp q_j(s_j)} \wedge 1$, the proposal code s_j is accepted to be the true code c_j. Here, $h_j \sharp \rho_j(s_j)$ and $h_j \sharp q_j(s_j)$ denote the probability of s_j under the pushforward of target model M and proposal model M_P's distributions, respectively. If s_j is not accepted, then we sample the true code c_j from the distribution $(h_j \sharp \rho_j - h_j \sharp q_j)_+$. Finally, we sample the next token from ρ_j conditional on the event that the next token hashes to the true code c_j.

To detect whether a sequence of tokens $y_{0:n}$ is watermarked, we reproduce the random seeds $s_{K:n}$ using the same proposal model M_P, pseudo-random generator PRG, and secret key sk. In the ideal case where the sequence is not perturbed, the seeds $s_{K:n}$ should be exactly the same as those in the generation phase since the same key and pseudo-random function are used. Then, we check whether the token y_j hashes to the code s_j, indicated by ξ_j. Finally, we say a subsequence $y_{K:n}$ is watermarked if the following condition holds:

$$\sum_{j=K}^{n} \xi_j \geq 1 - \alpha \text{ quantile of the random variable } \left(\sum_{j=K}^{n} Z_j\right), \tag{5.3}$$

where Z_j is Bernoulli random variable with expectation $h_j \sharp q_j(h_j(y_j))$, independently from each other. To find the $1-\alpha$ quantile, the probability mass function of $\sum_{j=K}^{n} Z_j$ can be

computed iteratively by dynamic programming:

$$\begin{aligned} &p_{1,1} = w_i,\ p_{1,0} = 1 - w_i \\ &p_{k+1,l} = w_{K+k} \cdot p_{k,l-1} + (1 - w_{K+k}) \cdot p_{k,l}, \end{aligned} \tag{5.4}$$

where $w_j = h_j \sharp q_j(h_j(y_j)) = \mathbb{P}(Z_j = 1)$ and $p_{k,-1} = 0, p_{k,l} = 0$ $(k < l)$ by default. The dynamic program computes the exact Poisson-binomial threshold under $\mathbf{H}_0$. Let $w_j = h_j \sharp q_j(h_j(y_j))$ and let $Z_j \sim \text{Bernoulli}(w_j)$ be independent for $i = K, \ldots, n$. The objective is to choose the smallest integer threshold

$$C\left(\alpha, \{w_j\}_{j=K}^n\right) = \min\left\{c : \mathbb{P}\left(\sum_{j=K}^n Z_j \geq c\right) \leq \alpha\right\},$$

so that any non-watermarked sequence crosses the threshold with probability at most α. It can be verified that $p_{k,l} = \mathbb{P}\left(\sum_{j=K}^{K+k-1} Z_j = l\right)$.

Alternatively, we may also use Bernoulli concentration inequalities to approximate the threshold:

$$\sum_{j=K}^n \xi_j \geq (1+\epsilon) \cdot \sum_{j=K}^n h_j \sharp q_j(h_j(y_j)) \tag{5.5}$$

where ϵ satisfies

$$(\epsilon - (1+\epsilon)\log(1+\epsilon)) \cdot \sum_{j=K}^n h_j \sharp q_j(h_j(y_j)) \leq \log \alpha.$$

In Table 5.1, we summarize the parameters used in our algorithm.

5.2.3 Theoretical guarantees

As a statistical watermarking scheme, SEAL enjoys formal statistical guarantees. One of the key properties of SEAL is that it is distortion-free when the bias b_{logit} is set to zero.

Theorem 5.2.1 (Distortion-freeness)**.** *When $b_{\text{logit}} = 0$, the watermark in Algorithm 5.2.1 is distortion-free. More precisely, for any sequence $t_{0:j-1}$ and any token w,*

$$\mathbb{P}_{\text{SEAL}}(t_j = w | p, t_{0:j-1}) = \rho_j(w | p, t_{0:j-1}),$$

where $\mathbb{P}_{\text{SEAL}}$ denotes the next-token probability under the SEAL generation phase, p is the prompt, and ρ_j represents the original model's next-token distribution.

This distortion-free guarantee ensures that the watermarked outputs have the same marginal distribution as the original model outputs, implying that the quality will not degrade after watermarking. Thus, SEAL is able to preserve the fidelity of the watermarked text relative to the original, unwatermarked output.

Parameter	Meaning
p	prompt
sk	secret key
K	attention window length
H	hash window length
KeyGen	(embedded) key generator
ρ_{j+1}	next token distribution
$\mathcal{S}$	EOS token set
τ	temperature of M
τ_P	temperature of M_P
b_{logit}	bias
M_P	smaller language model
M	target (watermarked) language model
PRG	pseudo-random generator
$\mathcal{H}$	family of hashing functions
α	Type I error

Table 5.1: Explanation of SEAL parameters.

Theorem 5.2.2 (False positive control)**.** *For any fixed (sub-)sequence y,*

$$\mathbb{P}_{\text{SEAL}}(\texttt{watermarked} = \text{True } \textit{for sequence } y) \leq \alpha,$$

where $\mathbb{P}_{\text{SEAL}}$ denotes the randomness in SEAL detection phase.

This Type I error guarantee controls the false positive error, ensuring that human-generated texts will only be mistakenly flagged as machine-generated with low probability.

When the proposal LLM is identical to the target LLM, the hash function is the identity mapping, and the attention window K is unlimited (consuming the prompt), in which case SEAL reduces to the statistically optimal watermark in Theorem 4.1.1, and therefore enjoys optimal Type II error. However, this reduction is not agnostic to the target LLM and the prompt. Therefore, in practice SEAL does not achieve the optimal Type II error, but still improves the statistical efficiency relative to existing work as shown in the next section.

Algorithm 5.2.1 SEAL Generation

procedure GENERATION(p, τ, τ_P, b_{logit}, $M(\cdot,\cdot)$, $M_P(\cdot,\cdot)$, sk, PRG$(\cdot,\cdot)$, K, H, EOS, $\mathcal{H}$)
▷ Generate a watermarked sequence of tokens $t = t_0 \dots t_j$
  $j \leftarrow 0$
  while True **do**
    $q_j \leftarrow M_P(t_{j-K:j-1}, \tau_P)$ ▷ Distribution of the j-th token according to M_P
    $\rho_j \leftarrow M(p \parallel t_{0:j-1}, \tau)$ ▷ Distribution of the j-th token according to M
    $key \leftarrow$ KEYGEN$(t_{j-H:j-1}; sk)$ ▷ Compute key by embedded key generator
    Sample h_j from $\mathcal{H}$ under key ▷ Select a random hash function
    $s_j \leftarrow$ PRG$(h_j \sharp q_j, key)$ ▷ Sample a random seed from $h_j \sharp q_j$ under key key
    $\rho_j(\omega) \leftarrow \rho_j(\omega) \cdot e^{\mathbb{1}(h_j(\omega)=s_j)\cdot b_{\text{logit}}/\tau}/Z, \ \forall \omega \in \Omega_0$ ▷ Add bias
    Sample $u \sim \text{Unif}(0,1)$
    if $u < \frac{h_j \sharp \rho_j(s_j)}{h_j \sharp q_j(s_j)}$ **then** ▷ Speculative decoding
      $c_j \leftarrow s_j$
    else
      Sample $c_j \propto (h_j \sharp \rho_j - h_j \sharp q_j)_+$; ▷ Maintain distortion-free
    Sample t_j from $\rho_j(\cdot | h(t_j) = c_j)$ ▷ Conditional on the hash code
    if $t_j \notin$ EOS **then**
      $j \leftarrow j + 1$
    else
      break;
  Return $t_{0:j}$

5.3 Experiments

In this section, we present our experimental setup and results. We evaluate the performance of our watermarking scheme by comparing it with several baselines.

5.3.1 Setup

We select Llama2-7B-chat [189] as the model to be watermarked and Phi-3-mini-128k-instruct [128] as the proposal model. We evaluate our watermark using the MARKMYWORDS benchmark [152].

MARKMYWORDS is an open-source benchmark designed to evaluate symmetric key watermarking schemes. It measures efficiency (watermark strength), quality (impact on utility), and tamper-resistance (ability to withstand simple perturbations without quality degradation). It has been used to benchmark multiple prior schemes applied to Llama2-7B-chat [189] and Mistral-7b [88]. Mark My Words performs a grid search over watermarking parameters. It selects the set of parameters with the best efficiency, defined as the number of tokens needed to detect the watermark at a fixed p-value (0.02 in the original paper) while preserving the original model's generation quality.

Algorithm 5.2.2 SEAL Detection

procedure DETECTION(y, τ_P, $M_P(\cdot,\cdot)$, α, sk, PRG$(\cdot,\cdot)$, $\mathcal{H}$, EOS, K, H)
▷ Detect whether a given sequence y is watermarked.
  $j \leftarrow K$
  while $y_j \notin$ EOS **do**
    $q_j \leftarrow M_P(y_{j-K:j-1}, \tau_P)$ ▷ Recompute the probability from M_P
    $key \leftarrow$ KEYGEN$(y_{j-H:j-1}; sk)$ ▷ Compute key by embedded key generator
    Sample h_j from $\mathcal{H}$ under key ▷ Select a random hash function
    $s_j \leftarrow$ PRG$(h_j \sharp q_j, key)$ ▷ Reconstruct the random seed
    $\xi_j \leftarrow \mathbb{1}\left(h_j(y_j) = s_j\right)$ ▷ Indicator of if y_j hashes to s_j
    $j \leftarrow j + 1$
  if Eq. (5.3) holds **then**
    `watermarked` = True for $y_{K:n}$ ▷ Detect if $y_{K:n}$ has many hash collisions
  else
    `watermarked` = False for $y_{K:n}$
  Return `watermarked`

Dataset. MARKMYWORDS generates 300 outputs spanning three tasks—book summarization, creative writing, and news article generation—which mimic potential misuse scenarios. Outputs are truncated after 1000 tokens.

Perturbations. Watermarked outputs undergo a set of transformations: (1) character level perturbations (contractions, expansions, misspellings and typos); (2) word level perturbations (random removal, addition, and swap of words in each sentence, replacing words with synonyms); (3) text-level perturbations (paraphrasing, translating the output to another language and back).

Baselines. This section evaluates four watermarking schemes: Distribution Shift [99], Exponential [3], Binary [38], and Inverse Transform [102].

5.3.2 Comparison to baselines

In this section, we present a comprehensive comparison of SEAL against these baselines in terms of quality, size, and tamper-resistance. We provide here a brief overview of each metric—detailed descriptions of which can be found in Section IV of Piet et al. [152].

1. **Quality** measures the utility of the watermarked text. It is computed using Llama-3 [124] with greedy decoding as a judge model. Quality scores range from 0 to 1.

2. **Size** represents the median number of tokens required to detect the watermark at a given p-value. All experiments enforce a Type I error constraint of $\alpha = 0.02$. Lower values indicate higher efficiency.

3. **Tamper-resistance** quantifies the resilience of the watermark under simple perturbations. It is measured by the normalized area under the curve (AUC) of the watermark success rate versus generation quality under different perturbations. This excludes more successful but expensive attacks such as paraphrasing, which we analyze separately in Section 5.3.3.

Table 5.2 shows the metrics of SEAL and the baseline watermarking schemes at different sampling temperature settings $\tau = 0.3, 0.7, 1.0$ of the target LLM (see Eq. (5.1) for the definition of τ). We do not report results for greedy decoding ($\tau = 0$): one special case of SEAL is equivalent to distribution shift in this setting, which is the only functional method at this temperature. SEAL is competitive across all metrics, particularly in terms of efficiency (size) and tamper-resistance. Although SEAL's quality is marginally lower than that of some baselines (e.g., inverse transform and binary), it remains within an acceptable range and close to distribution shift and exponential. SEAL excels in token efficiency, particularly at higher temperatures. At temperature $\tau = 1.0$ and $\tau = 0.7$, SEAL requires the fewest tokens to detect the watermark, outperforming all other methods, including distribution-shift and exponential. Even at lower temperatures ($\tau = 0.3$), SEAL remains highly competitive, requiring only a few more tokens than distribution shift while providing maximal tamper-resistance. Unlike the baselines, SEAL's parameters are not specifically tuned for efficiency, suggesting that further gains could be achieved with parameter optimization. Notably, SEAL demonstrates optimal tamper-resistance of 1.0 across all temperatures, outperforming baseline schemes by a large margin.

5.3.3 Tamper-resistance to paraphrase attacks

Paraphrase attacks rewrite model outputs using non-watermarked LLMs. Although more expensive, these pose a realistic threat given the increasing number of competitive open-sourced language models. As such, it is important to evaluate watermarking scheme's robustness against these attacks [100, 228].

Paraphrasing using GPT-3.5 is a particularly effective attack, removing most watermarks from the schemes in Piet et al. [152]. We found SEAL to be more robust against this attack than its predecessors.

Table 5.3 shows the proportion of outputs still watermarked after a paraphrase attack across different temperature settings ($\tau = 0.3, 0.7, 1.0$), for SEAL and the two most tamper-resistant schemes from Piet et al. [152]. Across all temperatures, SEAL significantly outperforms all baseline methods. This improvement can be attributed to SEAL's use of semantic instead of syntactic information from preceding tokens to set the random seed, making the watermark more resilient to paraphrasing. Additionally, the ability of distribution-shift to survive paraphrasing attacks varies more sharply than SEAL as temperature increases: SEAL demonstrates strong tamper-resistance, particularly against paraphrasing attacks.

Temp.	Scheme	Quality (↑)	Size (↓)	Tamper-resistance (↑)
0.3	Exponential	0.899 ± 0.001	∞	0.060 ± 0.040
	Inverse Transform	$\mathbf{0.910 \pm 0.001}$	∞	0.000 ± 0.000
	Binary	0.905 ± 0.007	∞	0.028 ± 0.014
	Distribution Shift	0.894 ± 0.002	$\mathbf{47.5 \pm 3.5}$	0.654 ± 0.007
	SEAL	0.893 ± 0.004	57.5 ± 2.5	$\mathbf{1.000 \pm 0.000}$
0.7	Exponential	0.901 ± 0.001	178.8 ± 11.0	0.414 ± 0.028
	Inverse Transform	0.908 ± 0.002	524.5 ± 24.7	0.194 ± 0.006
	Binary	$\mathbf{0.913 \pm 0.002}$	∞	0.298 ± 0.003
	Distribution Shift	0.894 ± 0.005	53.0 ± 2.8	$\mathbf{1.000 \pm 0.000}$
	SEAL	0.885 ± 0.003	$\mathbf{45.3 \pm 1.3}$	$\mathbf{1.000 \pm 0.000}$
1.0	Exponential	0.893 ± 0.002	89.2 ± 2.0	0.807 ± 0.064
	Inverse Transform	$\mathbf{0.908 \pm 0.002}$	201.8 ± 6.0	0.444 ± 0.138
	Binary	0.906 ± 0.001	391.3 ± 8.8	0.472 ± 0.030
	Distribution Shift	0.893 ± 0.005	109.0 ± 2.8	0.756 ± 0.004
	SEAL	0.873 ± 0.007	$\mathbf{73.3 \pm 1.5}$	$\mathbf{1.000 \pm 0.000}$

Table 5.2: Comparison of SEAL with baselines across quality, size, and tamper-resistance at different temperature settings. We compute the mean and standard deviation over different private keys. The best result in each category is highlighted in bold. SEAL demonstrates high efficiency and tamper-resistance. Its tamper-resistance consistently outperforms baselines and its efficiency outperforms baselines at higher temperatures.

Scheme / Temperature	0.3	0.7	1
Exponential	0.015 ± 0.007	0.054 ± 0.007	0.152 ± 0.035
Distribution Shift	0.093 ± 0.007	0.250 ± 0.035	0.113 ± 0.007
SEAL	$\mathbf{0.372 \pm 0.028}$	$\mathbf{0.480 \pm 0.069}$	$\mathbf{0.314 \pm 0.014}$

Table 5.3: Proportion of benchmark outputs still watermarked after paraphrasing, for SEAL, distribution-shift and exponential schemes. SEAL consistently achieves high tamper-resistance and significantly outperforms the baselines at all temperatures.

5.4 Conclusion

In this chapter, we have advanced the understanding of watermarking in large language models (LLMs) through practical algorithm design. Building on the theoretical insights in Chapter 4, we introduced SEAL (Semantic-aware Speculative Sampling), a novel watermarking algorithm that achieves both high efficiency and robustness. SEAL leverages semantic-aware random seeds, making it more resilient to paraphrase attacks compared to earlier methods. Additionally, SEAL's use of maximal coupling via speculative sampling allows it

to achieve greater efficiency, particularly at higher temperatures. These advantages make SEAL especially well-suited for practical, real-world applications where the persistence of watermarks under adversarial conditions is crucial.

While SEAL shows promising results for watermarking LLM outputs, SEAL's watermarking approach involves inference on a smaller model, which can slow down the overall inference and detection speed. This opens up several new directions for future study, including embedding statistical watermarks in advanced speculative decoding techniques and theoretical guarantees against adversaries.

Chapter 6

Anytime-Valid Statistical Watermarking

This chapter addresses the reliance on fixed-horizon statistical watermarking by developing the first e-value-based watermarking framework, *Anchored E-Watermarking*, that unifies optimal sampling with anytime-valid inference and valid early stopping. Unlike traditional approaches where optional stopping invalidates Type-I error guarantees, the framework enables valid, anytime-inference by constructing a test supermartingale for the detection process. By leveraging an anchor distribution to approximate the target model, this chapter characterizes the optimal e-value with respect to the *worst-case log-growth rate* and derives the optimal expected stopping time. The theoretical claims are substantiated by simulations and evaluations on established benchmarks, showing that the framework can significantly enhance sample efficiency, reducing the average token budget required for detection by 13–15% relative to state-of-the-art baselines.

6.1 Introduction

Despite the successes of statistical watermarking, state-of-the-art schemes remain constrained by two fundamental limitations. First, the design of generation and detection schemes lacks a unifying guiding principle. SEAL utilizes a hashed green-red-list seed heuristic adapted from Kirchenbauer et al. [99], but lacks rigorous justification. This calls for a characterization of optimal generation and detection schemes under given anchor distributions. Second, there exists a methodological disconnect between generation and detection. While text generation is inherently autoregressive and variable in length, current detection paradigms are *fixed-horizon* and *ad hoc*. In practice, for example, a detector may wish to process a stream of tokens and stop as soon as confidence is high. However, with the standard statistical paradigm based on p-values, it is not allowed to continuously monitor the test statistic and decide when to stop based on the current value—this is known as "p-hacking" and it invalidates the Type-I error guarantee. This inability to perform *post-hoc* sequential analysis severely limits detection

efficiency. These gaps highlight a critical research challenge:

How can we design provably efficient watermarking schemes leveraging anchor distributions that allow for valid early stopping?

In this chapter, we provide the first formal resolution to this question. We formulate the problem of *Anchored E-Watermarking*, where both the generator and detector share access to an anchor distribution p_0, aiming to watermark a family of distributions in the δ-proximity of p_0. Departing from classical p-value analysis, we adopt *e-values* as the central detection paradigm. E-values [171, 192, 193] are nonnegative random variables E satisfying $\mathbb{E}_{\mathbf{H}_0}[E] \leq 1$ under the null. Unlike p-values, e-values arise from supermartingales, and it is therefore possible to preserve Type-I error guarantees under optional stopping based on ongoing analysis of data. We analyze the optimal e-value for watermark detection with respect to the worst-case log-growth rate [94] and the expected stopping time [68, 202], thus fully characterize the average per-step growth rate of evidence and sample efficiency. Our main theoretical contributions are summarized informally as follows:

Theorem 6.1.1 (Informal version of Theorem 6.4.1 and Theorem 6.4.3)**.** *The optimal worst-case log-growth rate* $\mathbb{E}_{\mathbf{H}_1}[\log E]$ *under the alternative* $\mathbf{H}_1$ *is given by:*

$$J^* = h + \left(1 - \frac{\delta}{2}\right) \log \left(1 - \frac{\delta}{2}\right) + \frac{\delta}{2} \log \left(\frac{\delta}{2(n-1)}\right),$$

where $h = H(p_0)$ *is the Shannon entropy of the anchor distribution* p_0*,* n *is the cardinality of the sample space, and* $\delta > 0$ *is a robustness tolerance parameter. Furthermore, the optimal expected stopping time to achieve a Type-I error* α *scales as* $\frac{\log(1/\alpha)}{J^*}$*.*

To the best of our knowledge, this result presents the first application of e-values to the domain of statistical watermarking. By enabling valid sequential testing, our framework significantly improves detection efficiency, allowing the system to flag machine-generated text with fewer tokens than fixed-horizon counterparts. With the ability to early-stop, this sequential paradigm enhances robustness against adaptive attacks: because the detector can terminate immediately upon accumulating sufficient evidence, the watermark remains effective even if the attacker perturbs the text heavily in later segments (post-stopping). Consequently, our approach offers a theoretically rigorous and practically superior alternative to existing heuristic detection methods. Through experiments on real watermarking benchmark [152], we show that the theoretical scheme implied by Theorem 6.1.1 achieves consistently higher sample efficiency than state-of-the-art methods, reducing the token consumption by 13-15% across various temperature settings.

6.2 From Fixed-Horizon Watermarking to Anytime Detection

In this section, we give a recap of the statistical watermarking framework from Chapters 4 and 5 and set up the notations used in this chapter.

Statistical watermarking as a hypothesis testing problem. Formally, statistical watermarking is cast as a hypothesis testing problem. Let q denote the target distribution over the sample space $\mathcal{V}$, and let $\mathcal{S}$ represent the space of pseudorandom seeds. The watermarking protocol modifies the standard decoding mechanism of the LLM such that the output tokens $V \in \mathcal{V}$ and the pseudorandom seeds $S \in \mathcal{S}$ are sampled jointly from a watermarked distribution $\mathcal{P}_W$.

The detection task seeks to determine whether an observed sequence $v^{1:T} = (v^1, \ldots, v^T)$ was generated by the watermarked model. We write the sequential detector as $\mathcal{D}_T : \mathcal{V}^T \times \mathcal{S}^T \to \{\text{Watermarked}, \text{Unwatermarked}\}$. At each time t, the detector reconstructs the seed s^t from the secret key and computes a one-step score; the final decision is based on the accumulated evidence across the sequence. This reduces to testing for independence between the token sequence $v^{1:T}$ and the reconstructed seed sequence $s^{1:T}$. We define the null and alternative hypotheses following Chapter 4:

- **Null hypothesis $\mathbf{H}_0$:** The sequence $v^{1:T}$ is generated independently of the seeds $s^{1:T}$ (e.g., by a human or an unwatermarked model).
- **Alternative hypothesis $\mathbf{H}_1$:** The token-seed pairs $(v^t, s^t)_{t=1}^T$ are sampled from the joint watermarked mechanism, implying they come from the watermarked model.

It is important to distinguish this *distributional* watermarking approach from classical instance-level watermarking techniques applied to static media, such as images or audio [41]. While classical methods embed a signature into a specific, fixed outcome (post-generation), statistical watermarking modifies the stochastic sampling process itself, ensuring that any realization from the model carries the statistical evidence of its origin without requiring rigid alterations to the final output.

Statistical guarantees. A robust watermarking scheme is characterized by its ability to provide three fundamental guarantees.

First, to preserve generation quality and ensure watermarked content remains indistinguishable from ordinary samples, the distance (e.g., KL-divergence) between the watermarked distribution $\mathcal{P}_W$ and the original target distribution q must be minimized. Ideally, we seek a scheme that introduces zero distributional shift [77].

Definition 6.2.1 (Distortion-free)**.** A watermark is considered *distortion-free* (or unbiased) if the outcome marginal of the watermarked distribution $\mathcal{P}_W$ matches the target distribution

q. Formally, a distortion-free watermark satisfies:

$$\mathcal{P}_W(A \times \mathcal{S}) = q(A), \qquad \forall A \subseteq \mathcal{V}.$$

Second, it is often required in practice that the detector operates without access to the watermarked distribution since the underlying model's parameters are generally unknown to the detector. This means that the detection scheme needs to be designed for a *family of target distributions* in a model-agnostic way [102], leading to the next concept.

Definition 6.2.2 (Model-agnosticity)**.** A watermark is *model-agnostic* if the seed-marginal of $\mathcal{P}_W$ is chosen independently of the target distribution q. That is, the distribution of seeds S does not depend on the model's logits.

Third, the reliability of the watermark is quantified by standard statistical error rates. The *Type I error* (false positive rate) measures the probability of incorrectly identifying non-watermarked text (e.g., human-written) as watermarked. Given the high stakes of false accusations (e.g., in academic integrity), this error must be bounded by a small significance level α:

$$\mathbb{P}_{H_0}\Big(\mathcal{D}_T(V^{1:T}, S^{1:T}) = \text{Watermarked}\Big) \leq \alpha.$$

The *Type II error* (false negative rate) measures the probability that watermarked text fails to be detected. Minimizing this error is equivalent to maximizing the statistical power of the test. For a target error rate β, we require:

$$\mathbb{P}_{H_1}\Big(\mathcal{D}_T(V^{1:T}, S^{1:T}) = \text{Unwatermarked}\Big) \leq \beta.$$

6.2.1 Sequential hypothesis testing and e-values

Consider a measurable space $(\Omega, \mathcal{F})$ and a family of probability distributions $\mathcal{P}$. We are interested in testing a null hypothesis $\mathcal{H}_0 : P \in \mathcal{P}_0 \subset \mathcal{P}$ against an alternative $\mathcal{H}_1 : P \in \mathcal{P}_1 = \mathcal{P} \setminus \mathcal{P}_0$. An e-value is a random variable whose expectation is bounded by unity under the null hypothesis [192].

Definition 6.2.3 (E-value)**.** A nonnegative random variable E is an e-value for $\mathcal{H}_0$ if for all $P \in \mathcal{P}_0$,

$$\mathbb{E}_P[E] \leq 1.$$

Intuitively, an e-value represents the amount of wealth a gambler would have after betting against the null hypothesis, in a game where the game is fair or unfavorable under $\mathcal{H}_0$, and starting with an initial wealth of one. On the other hand, if E becomes large, this suggests that the null hypothesis is unlikely to be true.

Sequential evidence and stopping time guarantees. The primary advantage of e-values lies in their behavior regarding stopping times. In a sequential setting, we are given a filtration $(\mathcal{F}_t)_{t\geq 0}$ and construct a sequence of e-values $(E_t)_{t\geq 0}$. If $(E_t)_{t\geq 0}$ is a nonnegative supermartingale with respect to the null distributions (often called a *test martingale*), it allows for *anytime-valid* inference. This property derives from Ville's inequality, a time-uniform generalization of Markov's inequality.

Theorem 6.2.4 (Ville's inequality)**.** *Let $(E_t)_{t\geq 0}$ be a nonnegative supermartingale with respect to $P \in \mathcal{P}_0$ such that $E_0 \leq 1$. Then for any $\alpha \in (0,1)$,*

$$\mathbb{P}\left(\exists t \geq 0 : E_t \geq \frac{1}{\alpha}\right) \leq \alpha.$$

This result guarantees that a researcher can track the e-value process continuously and stop at any data-dependent time τ (a stopping time) while maintaining Type-I error control. Specifically, $\mathbb{E}_P[E_\tau] \leq 1$ holds for any stopping time τ (possibly unbounded), a kind of guarantee which is not available for p-values.

6.3 Problem Formulation

In this section, we formulate the problem of *Anchored E-Watermarking*. Similar to statistical watermarking, the goal is to embed statistical signals into samples from a target distribution to enable detection. However, in Anchored E-Watermarking, the generator and the detector share access to an anchor distribution that serves as a robust *a priori* estimate of the target distribution. This setting is common in practice and used in Chapter 5; for example, when watermarking Large Language Models or image generators, the target distribution is known to be human language or natural images, respectively. Consequently, watermarking efficacy can be improved by leveraging this structure. Besides, Anchored E-Watermarking uses e-values for detection, thus enjoying improved efficiency in sequential testing. In the following, we introduce the specific components of Anchored E-Watermarking.

Anchor distribution. In the Anchored E-Watermarking framework, we assume that both the generator and the detector share access to an *anchor distribution* $p_0 \in \Delta(\mathcal{V})$, which is in the same probability simplex as the target distribution q. This p_0 serves as the best *a priori* estimate of the target distribution. For instance, in the context of watermarking Large Language Models (LLMs), p_0 could be an open-source, smaller-scale LLM such as Qwen3-8B [212]. Leveraging its role as a known reference, we designate p_0 as the marginal distribution for the watermark signal (or random seed), denoted as s. Therefore, the seed space $\mathcal{S}$ is equal to $\mathcal{V}$ in our framework. Throughout the chapter, we assume p_0 satisfies the condition $\inf_{v\in\mathcal{V}} p_0(v) > \delta$[1] for a positive robustness tolerance parameter δ. This condition ensures p_0

[1]Here $\mathcal{V}$ denotes the finite alphabet on which the watermark is actually applied, not necessarily the raw tokenizer vocabulary. Following the green-red list scheme [99], raw tokens are mapped to a small number of

has enough randomness as required for watermarking [2] and that the δ-neighborhood defined below is a strict subset of the probability simplex.

Target distribution. The target distribution $q \in \Delta(\mathcal{V})$ represents the desired distribution from which (watermarked) output samples v are generated. Consistent with the definition of the anchor, we premise that q remains sufficiently proximal to p_0. Formally, we assume q resides within the δ-neighborhood of the anchor:

$$\mathcal{Q}(p_0, \delta) = \{q \in \Delta(\mathcal{V}) : \|q - p_0\|_1 \leq \delta\}.$$

Here $\|\cdot\|_1$ is the ℓ_1 distance and the radius $\delta > 0$ represents a robustness tolerance parameter that controls how the true target distribution q can deviate from the anchor. This assumption holds for most practical statistical watermarking scenarios; for example, in the LLM domain, high-performing models tend to converge towards the same underlying distribution of natural language, resulting in low statistical divergence between them.

Generator. The objective of the generator is to sample an output v from the target distribution q while embedding a statistical signal s. To achieve this, the generator constructs a joint distribution (or coupling) w over v and s, subject to the constraint that the marginals recover the target and anchor distributions, respectively. The feasible set of valid couplings is defined as:

$$\mathcal{P}(p_0, q) = \left\{ w \in \Delta(\mathcal{V} \times \mathcal{V}) : \sum_{s \in \mathcal{V}} w(v, s) = q(v), \quad \sum_{v \in \mathcal{V}} w(v, s) = p_0(s) \right\}.$$

Note that the generator has access to q, as is common in practice (e.g., when the generator is a service provider). During generation, the system samples the outcome v and the signal s jointly from w. This procedure ensures that the marginal distribution of the outcome v is unbiased (satisfying Definition 6.2.1), while the watermark signal is embedded within the statistical dependency between v and s.

E-value. In the detection phase, the detector sequentially receives an outcome-signal pair (v, s) and computes an e-value $e(v, s) \geq 0$. The objective is to distinguish between the following one-step hypotheses:

$$\mathbf{H}_0 : v \sim q \text{ and } s \sim p_0 \text{ are sampled independently,}$$
$$\mathbf{H}_1 : (v, s) \text{ are sampled from the joint coupling } w.$$

The null distribution q is arbitrary; in particular, human text need not lie in $\mathcal{Q}(p_0, \delta)$. To guarantee Type-I error at most α, the e-value must have expectation at most one under every

partition cells, and p_0 and q are distributions over the partitions. Therefore, the partition-level condition $\inf_{v \in \mathcal{V}} p_0(v) > \delta$ is expected to hold in general.

such null:

$$\sup_{q\in\Delta(\mathcal{V})} \mathbb{E}_{v\sim q,\, s\sim p_0}[e(v,s)] \le 1. \tag{6.1}$$

The set $\mathcal{Q}(p_0,\delta)$ is used below for alternative-side power analysis, not for null validity. We let $\mathcal{E}$ denote the set of all nonnegative functions $e : \mathcal{V}\times\mathcal{V}\to\mathbb{R}$ satisfying Eq. (6.1).

The process above describes a single conditional generation step. In an autoregressive model, all distributions are conditional on the history $\mathcal{F}_{t-1}$: the anchor and target become $p_0^t(\cdot) = p_0(\cdot \mid \mathcal{F}_{t-1})$ and $q^t(\cdot) = q(\cdot \mid \mathcal{F}_{t-1})$, and the detector uses the conditional e-value $e_t(v^t, s^t)$ induced by p_0^t. Since $\sum_s p_0^t(s)e_t(v,s) \le 1$ holds conditionally for every v, the product of one-step e-values is a test supermartingale under $\mathbf{H}_0$.

6.3.1 Robust log-optimality

While the definition of an e-value ensures validity (safety) under the null, it does not guarantee power (growth) under the alternative. To reject $\mathbf{H}_0$ effectively, we desire the e-value to be large when the data is generated from the alternative distribution $\mathbf{H}_1$. The standard criterion for selecting an e-value is *log-optimality*, often referred to as the "Kelly criterion" in the finance literature [94]. This approach seeks to maximize the expected logarithmic growth rate of evidence against the null, under the alternative. For simple hypotheses where $\mathcal{P}_0 = \{P\}$ and $\mathcal{P}_1 = \{Q\}$, the likelihood ratio $e^* = \frac{dQ}{dP}$ is log-optimal, such that

$$\mathbb{E}_Q[\log e^*] \ge \mathbb{E}_Q[\log e]$$

for any valid e-value e.

In Anchored E-Watermarking, the alternative is defined by the joint coupling w constructed by the generator. However, since the target distribution q lies in a set $\mathcal{Q}(p_0,\delta)$ that is unknown to the detector (i.e., the designer of the e-value), we must optimize the worst-case log-growth rate over arbitrary $q \in \mathcal{Q}(p_0,\delta)$. This gives rise to the following robust log-optimality problem.

Problem 6.3.1 (Robust log-optimality)**.** An e-value e is *robust log-optimal* in Anchored E-Watermarking if it optimizes the following objective:

$$\sup_{e\in\mathcal{E}} \inf_{q\in\mathcal{Q}(p_0,\delta)} \sup_{w\in\mathcal{P}(p_0,q)} \sum_{v,s\in\mathcal{V}} w(v,s)\cdot\log e(v,s). \tag{6.2}$$

In this formulation, the objective $\sum w(v,s)\log e(v,s)$ represents the log-growth rate under the alternative. The inner maximization $\sup_{w\in\mathcal{P}(p_0,q)}$ reflects the generator's optimization of the coupling w given knowledge of q; the minimization $\inf_{q\in\mathcal{Q}(p_0,\delta)}$ enforces robustness against the worst-case target distribution in the family; and the outer maximization $\sup_{e\in\mathcal{E}}$ seeks the optimal detector without access to q subject to model-agnosticity (Definition 6.2.2).

Remark 6.3.2 (Relationship to growth rate optimality in the worst case (GROW) [68]). GROW studies the worst-case optimal expected capital growth rate under a composite null $\mathcal{P}_0$ and alternative $\mathcal{P}_1$:

$$\mathrm{GROW}(\mathcal{P}_1) = \sup_{E \in \mathcal{E}(\mathcal{P}_0)} \inf_{\theta \in \mathcal{P}_1} \mathbb{E}_{P_\theta}[\log E],$$

where $\mathcal{E}(\mathcal{P}_0)$ is the set of all valid e-values for the null. If the generator were fixed as a map $q \mapsto w_q$, then Eq. (6.2) would reduce to a GROW problem over the induced alternative family $\{w_q : q \in \mathcal{Q}(p_0, \delta)\}$. Our formulation generalizes GROW because the generator is optimized actively through the inner maximization over couplings.

6.3.2 Stopping time

While Problem 6.3.1 addresses the one-step growth rate, it remains to analyze the sample complexity of the framework. In the context of sequential hypothesis testing, sample efficiency is characterized by the stopping time $\tau_\alpha = \inf\{n \in \mathbb{Z}_+ : W_n \geq \log(1/\alpha)\}$ under the alternative, where

$$W_n = \sum_{t=1}^{n} \log e(v^t, s^t),$$

is the accumulated wealth process. This quantity represents the number of samples required to reject the null hypothesis. We consider a stochastic process $\mu(\mathcal{A}, \mathcal{G})$ governed by the interaction between an adversary $\mathcal{A}$ and a generator $\mathcal{G}$. At each step $t \in \mathbb{Z}_+$:

- The adversary $\mathcal{A}$ selects a conditional target distribution $q^t \in \mathcal{Q}(p_0^t, \delta)$ based on the e-value and the history $(v^1, s^1), \ldots, (v^{t-1}, s^{t-1})$;
- The generator $\mathcal{G}$ specifies the joint coupling $w^t \in \mathcal{P}(p_0^t, q^t)$ and draws $(v^t, s^t) \sim w^t$.

Due to the adaptive nature of the adversary, the outcome sequence $v^1, v^2, \ldots$ may be generated autoregressively, conditional on previous outcomes. This formulation extends the i.i.d. setting of Chapter 4 to arbitrary dependent distributions. In this setting, we write the stopping time as $\tau_\alpha(e)$ to highlight the dependence on the e-value e. In the context of sequential testing [21, 39, 93, 172, 202], the expected stopping time $\mathbb{E}[\tau_\alpha]$ typically scales linearly with $\log(1/\alpha)$, modulated by an information-theoretic quantity capturing the complexity of the testing problem. Therefore, we are explicitly interested in the following question:

Problem 6.3.3 (Sample efficiency)**.** For any e-value e satisfying the constraint in Eq. (6.1), define its *sample complexity* by

$$\mathrm{SC}(e) = \sup_{\mathcal{A}} \inf_{\mathcal{G}} \lim_{\alpha \to 0} \frac{\mathbb{E}_{\mu(\mathcal{A},\mathcal{G})}[\tau_\alpha(e)]}{\log(1/\alpha)}.$$

We are interested in identifying the e-value that minimizes this sample complexity.

This formulation is stronger than the classical expected stopping-time setting because the adversary may alter the conditional target distribution at each time step. Thus samples need not be i.i.d., which is essential for autoregressive language-model watermarking.

6.4 Theoretical Results

In this section, we answer the questions posed in the previous section. We first derive the optimal detector for the single-step decision problem and then extend this analysis to the sequential setting to characterize the fundamental limit of sample efficiency. Let n denote the cardinality of the sample space $\mathcal{V}$ and $\mathbf{e}_v$ denote the one-hot vector (i.e., Dirac measure) that assigns mass 1 to $v \in \mathcal{V}$.

6.4.1 Optimal log-growth rate

We begin by solving the robust log-optimality problem defined in Problem 6.3.1. The following theorem provides a closed-form solution for both the optimal e-value and the worst-case-optimal generator behavior.

Theorem 6.4.1 (Log-growth rate)**.** *The optimal e-value that solves Eq.* (6.2) *is*

$$e^*(v,s) = \begin{cases} \frac{1-\delta/2}{p_0(s)}, & s = v, \\ \frac{\delta}{2(n-1)p_0(s)}, & s \neq v, \end{cases} \tag{6.3}$$

where $n = |\mathcal{V}|$. In particular, the middle infimum over $q \in \mathcal{Q}(p_0, \delta)$ is attained at $q_{a,b} = p_0 + \frac{\delta}{2}(\mathbf{e}_a - \mathbf{e}_b)$, $(a \neq b)$, and its optimal generator for the inner maximization is given by the maximal coupling

$$w^*_{a,b} = \operatorname{diag}(p_0) + \frac{\delta}{2}\left(\mathbf{e}_a\mathbf{e}_b^\top - \mathbf{e}_b\mathbf{e}_b^\top\right). \tag{6.4}$$

The optimal robust log-growth rate is

$$J^* = H(p_0) + \left(1 - \frac{\delta}{2}\right)\log\left(1 - \frac{\delta}{2}\right) + \frac{\delta}{2}\log\left(\frac{\delta}{2(n-1)}\right). \tag{6.5}$$

The result in Equation (6.5) fundamentally expresses the optimal worst-case growth rate of e-values in Anchored E-Watermarking. The first term is the entropy of the anchor distribution, which formalizes the intuition that watermarking is easier for higher-entropy anchors. The remaining terms equal $-H(\nu_\delta)$, where $\nu_\delta = (1-\delta/2, \delta/(2(n-1)), \ldots, \delta/(2(n-1))) \in \Delta([n])$. Thus the rate can be written as $H(p_0) - H(\nu_\delta)$, an entropy-gap analogue of the mutual information of a symmetric additive-noise channel.

Remark 6.4.2 (Relationship to SEAL). SEAL uses a smaller language model as p_0 and speculative decoding as the generator. For the optimal detector e^*, the generator-side inner problem is to maximize the diagonal mass of a coupling between q and p_0, which is exactly the maximal-coupling principle implemented by speculative decoding, confirming SEAL's choice of generator. On worst-case boundary points of $\mathcal{Q}(p_0, \delta)$ this reduces to the explicit coupling in Eq. (6.4). Our contribution relative to SEAL is therefore the optimal e-value detector and its anytime-valid sequential analysis.

6.4.2 Sample efficiency

Having characterized the one-step optimal growth rate, we now analyze the long-term performance of the watermark in a sequential setting. The following theorem connects the log-growth rate J^* to the sample complexity required to reject the null hypothesis against an adaptive adversary.

Theorem 6.4.3 (Expected stopping time)**.** *Let $\mu(\mathcal{A}, \mathcal{G})$ be the stochastic process in Problem 6.3.3, and let $\tau_\alpha(e)$ be the stopping time for e-value e. Let J^* be defined in Eq.* (6.5)*. Then:*

- ***Lower bound (converse):*** *For any valid e-value $e \in \mathcal{E}$,*

$$\sup_{\mathcal{A}} \inf_{\mathcal{G}} \liminf_{\alpha \downarrow 0} \frac{\mathbb{E}_{\mu(\mathcal{A},\mathcal{G})}[\tau_\alpha(e)]}{\log(1/\alpha)} \geq \frac{1}{J^*}.$$

- ***Upper bound (achievability):*** *For e^* in Theorem 6.4.1,*

$$\sup_{\mathcal{A}} \inf_{\mathcal{G}} \limsup_{\alpha \downarrow 0} \frac{\mathbb{E}_{\mu(\mathcal{A},\mathcal{G})}[\tau_\alpha(e^*)]}{\log(1/\alpha)} \leq \frac{1}{J^*}.$$

Consequently, e^* achieves the optimal first-order sample complexity $1/J^*$.

Theorem 6.4.3 establishes $1/J^*$ as the fundamental information-theoretic limit of first-order sample complexity for Anchored E-Watermarking. The optimal one-step e-value is therefore also optimal for sequential testing against an adaptive adversary, provided the conditional target distributions remain within the configured δ-neighborhood of the anchor.

Remark 6.4.4 (Relationship with the rates in Chapter 4). Chapter 4 shows that the minimum number of samples required to watermark with Type I error α scales as $\log(1/\alpha) \cdot \frac{\log(1/h)}{h}$, where h is the average entropy per token. While our rate $\frac{\log(1/\alpha)}{h}$ has the same scaling in terms of α, its dependence on the entropy h is improved. This is because Chapter 4 studies an asymptotic regime where $h \to 0$, while we fix an anchor distribution with lower bounded entropy arising from the condition $\inf_{v \in \mathcal{V}} p(v) > \delta$.

6.5 Experiments

6.5.1 Experiments on real data

We evaluate the performance of our watermarking scheme in Theorem 6.4.1 by comparing it with several baselines on a real watermarking task.

Setup. We select Llama2-7B-chat [189] with temperature 0.7 as target distribution (i.e., the model to be watermarked) and Phi-3-mini-128k-instruct [128] as the anchor distribution. We evaluate our watermark using the MarkMyWords benchmark [152].

Results. In this section, we present a comprehensive comparison of our method against these baselines in terms of quality and size. Quality measures the utility of the watermarked text. It is computed using Meta-Llama-3-8B-Instruct [124] with greedy decoding as a judge model, and ranges from zero to one. Size represents the median number of tokens required to detect the watermark at significance level α. All experiments enforce a Type-I error constraint of $\alpha = 0.02$. Lower values indicate higher efficiency. We compare against five state-of-the-art watermarking schemes: "Distribution Shift" [99], "Exponential" [3], "Binary" [38], "Inverse Transform" [102], and SEAL.

As shown in Table 6.1, our Anchored E-Watermarking framework significantly outperforms baseline methods in terms of detection efficiency while maintaining high generation quality. Specifically, our method improves over SEAL, confirming the superiority of the optimal e-value. Furthermore, we achieve nearly 2× speed improvement (from 145 to 72.0 tokens) compared to the best non-anchored baseline. Crucially, this efficiency gain does not come at the cost of text quality: the quality score of our method remains competitive with the baselines, demonstrating that E-value-based watermarking offers a superior trade-off between detectability and utility. Additional experiment results and hyperparameter setups are deferred to Appendix E.4.

Scheme	**Quality** (↑)	**Size** (↓)
Exponential	0.907	∞
Inverse Transform	0.917	734.0
Binary	**0.919**	∞
Distribution Shift	0.912	145.0
SEAL	0.901	84.5
Our e-value scheme	**0.919**	**72.0**

Table 6.1: Comparison of our e-value-based watermarking scheme in Theorem 6.4.1 with baselines across quality and size. We report the median over different private keys. The best result in each category is highlighted in bold. ∞ size suggests over half of the watermarked generations fail to be detected. E-value-based watermarking demonstrates higher efficiency compared with all baselines.

6.6 Conclusion

This chapter introduces *Anchored E-Watermarking*, an e-value-based watermarking framework that unifies optimal sampling with anytime-valid inference. Theorem 6.4.1 characterizes the optimal e-value e^* and the worst-case log-growth rate J^* in closed form, and Theorem 6.4.3 establishes a matching converse identifying $\log(1/\alpha)/J^*$ as the fundamental sample-complexity limit, even against an adaptive adversary that varies the target distribution within $\mathcal{Q}(p_0, \delta)$. On MarkMyWords, the resulting detector reduces the median detection size by 13–15% relative to the strongest baseline while preserving generation quality and Type-I error control.

Our framework has three principal limitations, each suggesting a direction for future work. First, the log-growth optimality of e^* requires the target to lie in the ℓ_1 ball $\mathcal{Q}(p_0, \delta)$ around a shared anchor; relaxing this to richer divergence balls (KL or Wasserstein) and to online-adapted anchors would broaden applicability. Second, the theory operates on a coarse partition of the vocabulary, leaving an open gap to the full-token regime in which $\inf_v p_0(v)$ may vanish. Third, our analysis targets sample efficiency rather than robustness against semantic attacks; coupling anytime-valid testing with explicit robustness guarantees, and extending the framework beyond text to image and code generation, remain natural next steps.

Bibliography

[1] Aakanksha Chowdhery et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113, 2023.

[2] Scott Aaronson. My AI safety lecture for UT effective altruism. Shtetl-Optimized, 2022. URL `https://scottaaronson.blog/?p=6823`. Blog post, November 29, 2022. Accessed: 2023-09-11.

[3] Scott Aaronson and Hendrik Kirchner. Watermarking GPT outputs, 2022. URL `https://www.scottaaronson.com/talks/watermark.ppt`. Slides, December 13, 2022.

[4] Sahar Abdelnabi and Mario Fritz. Adversarial watermarking transformer: Towards tracing text provenance with data hiding. In *2021 IEEE Symposium on Security and Privacy (SP)*, pages 121–140. IEEE, 2021.

[5] Adam Paszke et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32:8026–8037, 2019.

[6] Pranjal Aggarwal and Sean Welleck. L1: Controlling how long a reasoning model thinks with reinforcement learning. In *Second Conference on Language Modeling*, 2025. URL `https://openreview.net/forum?id=4jdIxXBNve`.

[7] Sudhanshu Agrawal, Risheek Garrepalli, Raghavv Goel, Christopher Lott, Fatih Porikli, and Mingu Lee. Structuring the future: Diffusion llm speculative decoding via calibrated draft graphs, 2026. URL `https://arxiv.org/abs/2509.18085`.

[8] 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.

[9] 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, editors, *Proceedings of 2nd Annual Workshop on Topology, Algebra, and Geometry in Machine Learning (TAG-ML)*, volume 221 of *Proceedings of Machine Learning Research*, pages 72–86. PMLR, 28 Jul 2023. URL `https://proceedings.mlr.press/v221/alberti23a.html`.

[10] Nima Anari, Ruiquan Gao, and Aviad Rubinstein. Parallel sampling via counting. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, pages 537–548, 2024.

[11] Anthropic. Claude 3 haiku: Our fastest model yet, March 2024. URL `https://www.anthropic.com/news/claude-3-haiku`. Accessed: 2026-06-11.

[12] Anthropic. Evaluating Claude's bioinformatics research capabilities with BioMysteryBench, April 2026. URL `https://www.anthropic.com/research/Evaluating-Claude-For-Bioinformatics-With-BioMysteryBench`. Accessed: 2026-06-11.

[13] Anthropic. Claude Code, 2026. URL `https://www.anthropic.com/claude-code`. Accessed: 2026-06-10.

[14] Marianne Arriola, Aaron Gokaslan, Justin T Chiu, Zhihan Yang, Zhixuan Qi, Jiaqi Han, Subham Sekhar Sahoo, and Volodymyr Kuleshov. Block diffusion: Interpolating between autoregressive and diffusion language models. *arXiv preprint arXiv:2503.09573*, 2025.

[15] Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 17981–17993. Curran Associates, Inc., 2021. URL `https://proceedings.neurips.cc/paper_files/paper/2021/file/958c530554f78bcd8e97125b70e6973d-Paper.pdf`.

[16] 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.

[17] Heli Ben-Hamu, Itai Gat, Daniel Severo, Niklas Nolte, and Brian Karrer. Accelerated sampling from masked diffusion models via entropy bounded unmasking. *arXiv preprint arXiv:2505.24857*, 2025.

[18] 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)*, pages 7096–7116, 2020.

[19] 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*, pages 455–475, 2020.

[20] 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.

[21] Leo Breiman. Optimal gambling systems for favorable games. *Fourth Berkeley Symposium on Probability and Statistics*, pages 65–78, 1961.

[22] 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.

[23] 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, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf`.

[24] Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, Harsha Nori, Hamid Palangi, Marco Tulio Ribeiro, and Yi Zhang. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*, 2023.

[25] Sébastien Bubeck, Christian Coester, Ronen Eldan, Timothy Gowers, Yin Tat Lee, Alexandru Lupsasca, Mehtaab Sawhney, Robert Scherrer, Mark Sellke, Brian K. Spears, Derya Unutmaz, Kevin Weil, Steven Yin, and Nikita Zhivotovskiy. Early science acceleration experiments with GPT-5, November 2025. URL `https://cdn.openai.com/pdf/4a25f921-e4e0-479a-9b38-5367b47e8fd0/early-science-acceleration-experiments-with-gpt-5.pdf`. Accessed: 2026-06-11.

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

[27] Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. Maskgit: Masked generative image transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11315–11325, 2022.

[28] Harrison Chase. LangChain. `https://github.com/langchain-ai/langchain`, 2022. Released: 2022-10-17.

[29] Boyuan Chen, Diego Martí Monsó, Yilun Du, Max Simchowitz, Russ Tedrake, and Vincent Sitzmann. Diffusion forcing: Next-token prediction meets full-sequence diffusion. *Advances in Neural Information Processing Systems*, 37:24081–24125, 2024.

[30] Canyu Chen and Kai Shu. Can llm-generated misinformation be detected? In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun, editors, *International Conference on Learning Representations*, volume 2024, pages 34687–34726, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/94bbcb744bbada8808fda05b9d9290d6-Paper-Conference.pdf.

[31] 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*, 2024. URL https://openreview.net/forum?id=VaXnxQ3UKo.

[32] 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.

[33] 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*, 2024. URL https://openreview.net/forum?id=m5106RRLgx.

[34] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code, 2021. URL https://arxiv.org/abs/2107.03374.

[35] Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. Do not think that much for 2+ 3=? on the overthinking of o1-like llms. *arXiv preprint arXiv:2412.21187*, 2024.

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

[37] Miranda Christ and Sam Gunn. Pseudorandom error-correcting codes. In *Annual International Cryptology Conference*, pages 325–347. Springer, 2024.

[38] Miranda Christ, Sam Gunn, and Or Zamir. Undetectable watermarks for language models. *arXiv preprint arXiv:2306.09194*, 2023.

[39] Ben Chugg, Santiago Cortes-Gomez, Bryan Wilder, and Aaditya Ramdas. Auditing fairness by betting. *Advances in Neural Information Processing Systems*, 36:6070–6091, 2023.

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

[41] Ingemar Cox, Matthew Miller, Jeffrey Bloom, Jessica Fridrich, and Ton Kalker. *Digital Watermarking and Steganography*. Morgan Kaufmann, Burlington, MA, 2 edition, 2007. ISBN 978-0-12-372585-1.

[42] Alejandro Cuadron, Dacheng Li, Wenjie Ma, Xingyao Wang, Yichuan Wang, Siyuan Zhuang, Shu Liu, Luis Gaspar Schroeder, Tian Xia, Huanzhi Mao, Nicholas Thumiger, Aditya Desai, Ion Stoica, Ana Klimovic, Graham Neubig, and Joseph E. Gonzalez. The danger of overthinking: Examining the reasoning-action dilemma in agentic tasks. *arXiv preprint arXiv:2502.08235*, 2025.

[43] Debarati Das, Karin De Langis, Anna Martin-Boyle, Jaehyung Kim, Minhwa Lee, Zae Myung Kim, Shirley Anugrah Hayati, Risako Owan, Bin Hu, Ritik Parkar, Ryan Koo, Jonginn Park, Aahan Tyagi, Libby Ferland, Sanjali Roy, Vincent Liu, and Dongyeop Kang. Under the surface: Tracking the artifactuality of llm-generated data. *arXiv preprint arXiv:2401.14698*, 2024.

[44] Google DeepMind. Gemini diffusion, 2025. URL `https://deepmind.google/models/gemini-diffusion/`.

[45] DeepSeek-AI. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, Sept 2025. ISSN 1476-4687. doi: 10.1038/s41586-025-09422-z. URL `http://dx.doi.org/10.1038/s41586-025-09422-z`.

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

[47] 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*, pages 5793–5831. PMLR, 2022.

[48] 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.

[49] Jaiden Fairoze, Sanjam Garg, Somesh Jha, Saeed Mahloujifar, Mohammad Mahmoody, and Mingyuan Wang. Publicly detectable watermarking for language models. *arXiv preprint arXiv:2310.18491*, 2023.

[50] 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.

[51] Guhao Feng, Yihan Geng, Jian Guan, Wei Wu, Liwei Wang, and Di He. Theoretical benefit and limitation of diffusion language model. *arXiv preprint arXiv:2502.09622*, 2025.

[52] Tony Feng, Junehyuk Jung, Sang-hyun Kim, Carlo Pagano, Sergei Gukov, Chiang-Chiang Tsai, David Woodruff, Adel Javanmard, Aryan Mokhtari, Dawsen Hwang, Yuri Chervonyi, Jonathan N. Lee, Garrett Bingham, Trieu H. Trinh, Vahab Mirrokni, Quoc V. Le, and Thang Luong. Aletheia tackles firstproof autonomously, 2026. URL `https://arxiv.org/abs/2602.21201`.

[53] Tony Feng, Trieu H. Trinh, Garrett Bingham, Dawsen Hwang, Yuri Chervonyi, Junehyuk Jung, Joonkyung Lee, Carlo Pagano, Sang-hyun Kim, Federico Pasqualotto, Sergei Gukov, Jonathan N. Lee, Junsu Kim, Kaiying Hou, Golnaz Ghiasi, Yi Tay, YaGuang Li, Chenkai Kuang, Yuan Liu, Hanzhao Lin, Evan Zheran Liu, Nigamaa Nayakanti, Xiaomeng Yang, Heng-Tze Cheng, Demis Hassabis, Koray Kavukcuoglu, Quoc V. Le, and Thang Luong. Towards autonomous mathematics research, 2026. URL `https://arxiv.org/abs/2602.10177`.

[54] Pierre Fernandez, Antoine Chaffin, Karim Tit, Vivien Chappelier, and Teddy Furon. Three bricks to consolidate watermarks for large language models. *arXiv preprint arXiv:2308.00113*, 2023.

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

[56] Hengyu Fu, Baihe Huang, Virginia Adams, Charles Wang, Junkeun Yi, Mohammad Mahdi Kamani, Venkat Srinivasan, and Jiantao Jiao. From bits to rounds: Parallel decoding with exploration for diffusion language models. In *Forty-third International Conference on Machine Learning*, 2026.

[57] Yu Fu, Deyi Xiong, and Yue Dong. Watermarking conditional text generation for ai detection: Unveiling challenges and a semantic-aware watermark remedy. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18003–18011, 2024.

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

[59] 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.

[60] Gemini Team, Google. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

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

[62] Noah Golowich and Ankur Moitra. Edit distance robust watermarks via indexing pseudorandom codes. *arXiv preprint arXiv:2406.02633*, 2024.

[63] Shansan Gong, Shivam Agarwal, Yizhe Zhang, Jiacheng Ye, Lin Zheng, Mukai Li, Chenxin An, Peilin Zhao, Wei Bi, Jiawei Han, Hao Peng, and Lingpeng Kong. Scaling diffusion language models via adaptation from autoregressive models. *arXiv preprint arXiv:2410.17891*, 2024.

[64] Google DeepMind. AI achieves silver-medal standard solving International Mathematical Olympiad problems, 2024. URL `https://deepmind.google/discover/blog/ai-solves-imo-problems-at-silver-medal-level/`. Accessed: 2026-06-10.

[65] Juraj Gottweis, Wei-Hung Weng, Alexander Daryin, Tao Tu, Petar Sirkovic, Artiom Myaskovsky, Grzegorz Glowaty, Felix Weissenberger, Alessio Orlandi, Dan Popovici, Anil Palepu, Keran Rong, Ryutaro Tanno, Khaled Saab, Fan Zhang, Jacob Blum, Andrew Carroll, Kavita Kulkarni, Nenad Tomasev, Dina Zverinski, Ivor Rendulic, Elahe Vedadi, Florian Hasler, Luka Rimanic, Marina Boia, Ivan Budiselic, Ben Feinstein, Mathias Bellaiche, Tom Sheffer, Jan Freyberg, Jeremy Ratcliff, Ottavia Bertolli, Katherine Chou, Avinatan Hassidim, Burak Gokturk, Amin Vahdat, Yuan Guan, Vikram Dhillon, Eeshit Dhaval Vaishnav, Byron Lee, Tiago R. D. Costa, José R. Penadés, Gary Peltz, Yossi Matias, James Manyika, Demis Hassabis, Yunhan Xu, Pushmeet Kohli, Annalisa Pawlosky, Alan Karthikesalingam, and Vivek Natarajan. Accelerating scientific discovery with Co-Scientist. *Nature*, May 2026. doi: 10.1038/s41586-026-10644-y. URL `https://doi.org/10.1038/s41586-026-10644-y`.

[66] 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`.

[67] Paul J. Goulart and Yuwen Chen. Clarabel: An interior-point solver for conic programs with quadratic objectives, 2024.

[68] Peter Grünwald, Rianne de Heide, and Wouter M Koolen. Safe testing. In *2020 Information Theory and Applications Workshop (ITA)*, pages 1–54. IEEE, 2020.

[69] Bradley Guo, Jingwen Gu, Jin Peng Zhou, and Wen Sun. Learning to self-correct through chain-of-thought verification. In *Second Workshop on Test-Time Adaptation: Putting Updates to the Test! at ICML 2025*, 2025.

[70] 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.

[71] 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.

[72] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

[73] 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, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=iBBcRUlOAPR.

[74] Abe Hou, Jingyu Zhang, Tianxing He, Yichen Wang, Yung-Sung Chuang, Hongwei Wang, Lingfeng Shen, Benjamin Van Durme, Daniel Khashabi, and Yulia Tsvetkov. SemStamp: A semantic watermark with paraphrastic robustness for text generation. In Kevin Duh, Helena Gomez, and Steven Bethard, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4067–4082, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.226. URL https://aclanthology.org/2024.naacl-long.226/.

[75] Abe Bohan Hou, Jingyu Zhang, Yichen Wang, Daniel Khashabi, and Tianxing He. k-semstamp: A clustering-based semantic watermark for detection of machine-generated text. *arXiv preprint arXiv:2402.11399*, 2024.

[76] Hengyuan Hu, Aniket Das, Dorsa Sadigh, and Nima Anari. Diffusion models are secretly exchangeable: Parallelizing ddpms via autospeculation, 2025. URL https://arxiv.org/abs/2505.03983.

[77] Zhengmian Hu, Lichang Chen, Xidong Wu, Yihan Wu, Hongyang Zhang, and Heng Huang. Unbiased watermark for large language models. *arXiv preprint arXiv:2310.10669*, 2023.

[78] 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*, 2024.

[79] Baihe Huang, Hanlin Zhu, Banghua Zhu, Kannan Ramchandran, Michael I Jordan, Jason D Lee, and Jiantao Jiao. Towards optimal statistical watermarking. *arXiv preprint arXiv:2312.07930*, 2023.

[80] Baihe Huang, Hanlin Zhu, Julien Piet, Banghua Zhu, Jason D Lee, Kannan Ramchandran, Michael Jordan, and Jiantao Jiao. Watermarking using semantic-aware speculative sampling: from theory to practice, 2025. URL `https://openreview.net/pdf?id=LdIlnsePNt`.

[81] Baihe Huang, Shanda Li, Tianhao Wu, Yiming Yang, Ameet Talwalkar, Kannan Ramchandran, Michael I. Jordan, and Jiantao Jiao. Sample complexity and representation ability of test-time scaling paradigms. In *The Fourteenth International Conference on Learning Representations*, 2026. URL `https://openreview.net/forum?id=RJDIX75TEE`.

[82] Baihe Huang, Eric Xu, Kannan Ramchandran, Jiantao Jiao, and Michael I Jordan. Towards anytime-valid statistical watermarking. *arXiv preprint arXiv:2602.17608*, 2026.

[83] Pengcheng Huang, Tianming Liu, Zhenghao Liu, Yukun Yan, Shuo Wang, Tong Xiao, Zulong Chen, and Maosong Sun. Empirical analysis of decoding biases in masked diffusion models, 2026. URL `https://arxiv.org/abs/2508.13021`.

[84] 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*, 2024. URL `https://openreview.net/forum?id=ayF8bEKYQy`.

[85] 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, 2023. URL `https://arxiv.org/abs/2303.06135`.

[86] Kevin Jamieson, Matthew Malloy, Robert Nowak, and Sébastien Bubeck. lil'ucb: An optimal exploration algorithm for multi-armed bandits. In *Conference on Learning Theory*, pages 423–439. PMLR, 2014.

[87] Adeeb M Jarrah, Yousef Wardat, and Patricia Fidalgo. Using ChatGPT in academic writing is (not) a form of plagiarism: What does the literature say. *Online Journal of Communication and Media Technologies*, 13(4):e202346, 2023.

[88] Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7B, 2023. Licensed under the Apache 2.0 license.

[89] Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. SWE-bench: Can language models resolve real-world GitHub issues?, 2023. URL `https://arxiv.org/abs/2310.06770`.

[90] 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.

[91] Nurul Shamimi Kamaruddin, Amirrudin Kamsin, Lip Yee Por, and Hameedur Rahman. A review of text watermarking: theory, methods, and applications. *IEEE Access*, 6: 8011–8028, 2018.

[92] 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.

[93] Emilie Kaufmann and Wouter M Koolen. Mixture martingales revisited with applications to sequential tests and confidence intervals. *Journal of Machine Learning Research*, 22(246):1–44, 2021.

[94] John L Kelly. A new interpretation of information rate. *The Bell System Technical Journal*, 35(4):917–926, 1956.

[95] Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, Heather Miller, Matei Zaharia, and Christopher Potts. DSPy: Compiling declarative language model calls into self-improving pipelines, 2023. URL `https://arxiv.org/abs/2310.03714`.

[96] Jaeyeon Kim, Kulin Shah, Vasilis Kontonis, Sham Kakade, and Sitan Chen. Train for the worst, plan for the best: Understanding token ordering in masked diffusions. *arXiv preprint arXiv:2502.06768*, 2025.

[97] Kimi. Kimi k1.5: Scaling reinforcement learning with llms, 2025. URL `https://arxiv.org/abs/2501.12599`.

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

[99] John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A watermark for large language models. *arXiv preprint arXiv:2301.10226*, 2023.

[100] John Kirchenbauer, Jonas Geiping, Yuxin Wen, Manli Shu, Khalid Saifullah, Kezhi Kong, Kasun Fernando, Aniruddha Saha, Micah Goldblum, and Tom Goldstein. On the reliability of watermarks for large language models. *arXiv preprint arXiv:2306.04634*, 2023.

[101] Ryuto Koike, Masahiro Kaneko, and Naoaki Okazaki. Outfox: Llm-generated essay detection through in-context learning with adversarially generated examples. *arXiv preprint arXiv:2307.11729*, 2023.

[102] Rohith Kuditipudi, John Thickstun, Tatsunori Hashimoto, and Percy Liang. Robust distortion-free watermarks for language models. *arXiv preprint arXiv:2307.15593*, 2023.

[103] Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, John D Co-Reyes, Avi Singh, Kate Baumli, Shariq Iqbal, Colton Bishop, Rebecca Roelofs, Lei M Zhang, Kay McKinney, Disha Shrivastava, Cosmin Paduraru, George Tucker, Doina Precup, Feryal Behbahani, and Aleksandra Faust. Training language models to self-correct via reinforcement learning. *arXiv preprint arXiv:2409.12917*, 2024.

[104] Thomas Kwa, Ben West, Joel Becker, Amy Deng, Katharyn Garcia, Max Hasin, Sami Jawhar, Megan Kinniment, Nate Rush, Sydney Von Arx, Ryan Bloom, Thomas Broadley, Haoxing Du, Brian Goodrich, Nikola Jurkovic, Luke Harold Miles, Seraphina Nix, Tao Lin, Neev Parikh, David Rein, Lucas Jun Koba Sato, Hjalmar Wijk, Daniel M. Ziegler, Elizabeth Barnes, and Lawrence Chan. Measuring AI ability to complete long software tasks, 2025. URL `https://arxiv.org/abs/2503.14499`.

[105] Inception Labs, Samar Khanna, Siddhant Kharbanda, Shufan Li, Harshit Varma, Eric Wang, Sawyer Birnbaum, Ziyang Luo, Yanis Miraoui, Akash Palrecha, Stefano Ermon, Aditya Grover, and Volodymyr Kuleshov. Mercury: Ultra-fast language models based on diffusion. *arXiv preprint arXiv:2506.17298*, 2025.

[106] LangChain AI. LangGraph. `https://github.com/langchain-ai/langgraph`, 2023. Official documentation: `https://docs.langchain.com/oss/python/langgraph/`.

[107] Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning*, pages 19274–19286. PMLR, 2023.

[108] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

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

[110] 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.

[111] Xiang Li, Feng Ruan, Huiyuan Wang, Qi Long, and Weijie J Su. A statistical framework of watermarks for large language models: Pivot, detection efficiency and optimal rules. *arXiv preprint arXiv:2404.01245*, 2024.

[112] 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*, 2024.

[113] Yujia Li, David Choi, Junyoung Chung, Nate Kushman, Julian Schrittwieser, Rémi Leblond, Tom Eccles, James Keeling, Felix Gimeno, Agustin Dal Lago, Thomas Hubert, Peter Choy, Cyprien de Masson d'Autume, Igor Babuschkin, Xinyun Chen, Po-Sen Huang, Johannes Welbl, Sven Gowal, Alexey Cherepanov, James Molloy, Daniel J. Mankowitz, Esme Sutherland Robson, Pushmeet Kohli, Nando de Freitas, Koray Kavukcuoglu, and Oriol Vinyals. Competition-level code generation with AlphaCode. *Science*, 378(6624):1092–1097, 2022. doi: 10.1126/science.abq1158. URL `https://doi.org/10.1126/science.abq1158`.

[114] 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*, 2024. URL `https://openreview.net/forum?id=3EWTEy9MTM`.

[115] Weixin Liang, Zachary Izzo, Yaohui Zhang, Haley Lepp, Hancheng Cao, Xuandong Zhao, Lingjiao Chen, Haotian Ye, Sheng Liu, Zhi Huang, Daniel A. McFarland, and James Y. Zou. Monitoring AI-modified content at scale: A case study on the impact of ChatGPT on AI conference peer reviews, 2024. URL `https://arxiv.org/abs/2403.07183`.

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

[117] 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.

[118] Qingwen Lin, Boyan Xu, Guimin Hu, Zijian Li, Zhifeng Hao, Keli Zhang, and Ruichu Cai. Cmcts: A constrained monte carlo tree search framework for mathematical reasoning in large language model, 2025. URL `https://arxiv.org/abs/2502.11169`.

[119] Aiwei Liu, Leyi Pan, Xuming Hu, Shu'ang Li, Lijie Wen, Irwin King, and Philip S Yu. An unforgeable publicly verifiable watermark for large language models. *arXiv preprint arXiv:2307.16230*, 2023.

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

[121] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023. URL `https://arxiv.org/abs/2304.08485`.

[122] Yepeng Liu and Yuheng Bu. Adaptive text watermark for large language models. *arXiv preprint arXiv:2401.13927*, 2024.

[123] Zhiyuan Liu, Yicun Yang, Yaojie Zhang, Junjie Chen, Chang Zou, Qingyuan Wei, Shaobo Wang, and Linfeng Zhang. dllm-cache: Accelerating diffusion large language models with adaptive caching. *arXiv preprint arXiv:2506.06295*, 2025.

[124] Llama Team, AI @ Meta. The llama 3 herd of models. *arXiv e-prints*, page arXiv preprint 2407.21783, 2024.

[125] Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. *arXiv preprint arXiv:2310.16834*, 2023.

[126] 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, editors, *Advances in Neural Information Processing Systems*, 2022. URL `https://openreview.net/forum?id=NQFFNdsOGD`.

[127] 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`.

[128] Marah Abdin et al. Phi-3 technical report: A highly capable language model locally on your phone, 2024. URL `https://arxiv.org/abs/2404.14219`.

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

[130] 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.

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

[132] Silvia Milano, Joshua A McGrane, and Sabina Leonelli. Large language models challenge the future of higher education. *Nature Machine Intelligence*, 5(4):333–334, 2023.

[133] Piotr Molenda, Adian Liusie, and Mark JF Gales. Waterjudge: Quality-detection trade-off when watermarking large language models. *arXiv preprint arXiv:2403.19548*, 2024.

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

[135] 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 Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Findings of the Association for Computational Linguistics: ACL 2025*, pages 25127–25152, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.1289. URL `https://aclanthology.org/2025.findings-acl.1289/`.

[136] Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. WebGPT: Browser-assisted question-answering with human feedback, 2022. URL `https://arxiv.org/abs/2112.09332`.

[137] Nelson Elhage et al. A mathematical framework for transformer circuits. *Transformer Circuits Thread*, 1:1, 2021.

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

[139] Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. *arXiv preprint arXiv:2502.09992*, 2025.

[140] Harsha Nori, Naoto Usuyama, Nicholas King, Scott Mayer McKinney, Xavier Fernandes, Sheng Zhang, and Eric Horvitz. From medprompt to o1: Exploration of run-time strategies for medical challenge problems and beyond, 2024. URL `https://arxiv.org/abs/2411.03590`.

[141] Alexander Novikov, Ngân Vũ, Marvin Eisenberger, Emilien Dupont, Po-Sen Huang, Adam Zsolt Wagner, Sergey Shirobokov, Borislav Kozlovskii, Francisco J. R. Ruiz, Abbas Mehrabian, M. Pawan Kumar, Abigail See, Swarat Chaudhuri, George Holland, Alex Davies, Sebastian Nowozin, Pushmeet Kohli, and Matej Balog. Alphaevolve: A coding agent for scientific and algorithmic discovery. *arXiv preprint arXiv:2506.13131*, 2025.

[142] OpenAI. GPT-4 technical report, 2023.

[143] OpenAI. Openai o3-mini, 2025. URL `https://openai.com/index/openai-o3-mini/`.

[144] OpenAI. Openai o1 system card, 2026. URL `https://arxiv.org/abs/2412.16720`.

[145] OpenAI. GPT Image 2 model, 2026. URL `https://developers.openai.com/api/docs/models/gpt-image-2`. Accessed: 2026-06-11.

[146] OpenAI. Introducing GPT-Rosalind for life sciences research, April 2026. URL `https://openai.com/index/introducing-gpt-rosalind/`. Accessed: 2026-06-11.

[147] OpenAI. Planar point sets with many unit distances, 2026. URL `https://cdn.openai.com/pdf/74c24085-19b0-4534-9c90-465b8e29ad73/unit-distance-proof.pdf`. Accessed: 2026-06-10.

[148] OpenClaw. OpenClaw – personal AI assistant, 2026. URL `https://openclaw.ai/`. Accessed: 2026-06-14.

[149] Jingyang Ou, Shen Nie, Kaiwen Xue, Fengqi Zhu, Jiacheng Sun, Zhenguo Li, and Chongxuan Li. Your absorbing discrete diffusion secretly models the conditional distributions of clean data. arXiv preprint arXiv:2406.03736, 2024.

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

[151] 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*, pages 40523–40550, 2024.

[152] Julien Piet, Chawin Sitawarin, Vivian Fang, Norman Mu, and David Wagner. Mark my words: Analyzing and evaluating language model watermarks. *arXiv preprint arXiv:2312.00273*, 2023.

[153] Mihir Prabhudesai, Mengning Wu, Amir Zadeh, Katerina Fragkiadaki, and Deepak Pathak. Diffusion beats autoregressive in data-constrained settings. *arXiv preprint arXiv:2507.15857*, 2025.

[154] 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.

[155] 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.

[156] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021. URL `https://arxiv.org/abs/2103.00020`.

[157] Naina Raisinghani. Nano Banana 2: Combining pro capabilities with lightning-fast speed, February 2026. URL `https://blog.google/innovation-and-ai/technology/ai/nano-banana-2/`. Accessed: 2026-06-11.

[158] Aaditya Ramdas, Peter Grünwald, Vladimir Vovk, and Glenn Shafer. Game-theoretic statistics and safe anytime-valid inference. *Statistical Science*, 38(4):576–601, 2023. doi: 10.1214/23-sts894.

[159] Jie Ren, Han Xu, Yiding Liu, Yingqian Cui, Shuaiqiang Wang, Dawei Yin, and Jiliang Tang. A robust semantics-based watermark for large language model against paraphrasing. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 613–625, Mexico City, Mexico, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-naacl.40. URL `https://aclanthology.org/2024.findings-naacl.40/`.

[160] Stefano Giovanni Rizzo, Flavio Bertini, and Danilo Montesi. Fine-grain watermarking for intellectual property protection. *EURASIP Journal on Information Security*, 2019: 1–20, 2019.

[161] Herbert E. Robbins. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 58(5):527–535, 1952.

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

[163] Rounak Saha, Gurusha Juneja, Dayita Chaudhuri, Naveeja Sajeevan, Nihar B. Shah, and Danish Pruthi. Policies permitting LLM use for polishing peer reviews are currently not enforceable, 2026. URL `https://arxiv.org/abs/2603.20450`.

[164] Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Mariano Marroquin, Justin T Chiu, Alexander M Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. *Advances in Neural Information Processing Systems*, 37:130136–130184, 2024. URL https://openreview.net/forum?id=L4uaAR4ArM.

[165] Ryoma Sato, Yuki Takezawa, Han Bao, Kenta Niwa, and Makoto Yamada. Embarrassingly simple text watermarks. *arXiv preprint arXiv:2310.08920*, 2023.

[166] Timo Schick, Jane Dwivedi-Yu, Roberto Dessi, Roberta Raileanu, Maria Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools, 2023. URL https://arxiv.org/abs/2302.04761.

[167] Andrew Sellergren, Sahar Kazemzadeh, Tiam Jaroensri, Atilla Kiraly, Madeleine Traverse, Timo Kohlberger, Shawn Xu, Fayaz Jamil, Cían Hughes, Charles Lau, Justin Chen, Fereshteh Mahvar, Liron Yatziv, Tiffany Chen, Bram Sterling, Stefanie Anna Baby, Susanna Maria Baby, Jeremy Lai, Samuel Schmidgall, Lu Yang, Kejia Chen, Per Bjornsson, Shashir Reddy, Ryan Brush, Kenneth Philbrick, Mercy Asiedu, Ines Mezerreg, Howard Hu, Howard Yang, Richa Tiwari, Sunny Jansen, Preeti Singh, Yun Liu, Shekoofeh Azizi, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Riviere, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean-bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Elena Buchatskaya, Jean-Baptiste Alayrac, Dmitry Lepikhin, Vlad Feinberg, Sebastian Borgeaud, Alek Andreev, Cassidy Hardin, Robert Dadashi, Léonard Hussenot, Armand Joulin, Olivier Bachem, Yossi Matias, Katherine Chou, Avinatan Hassidim, Kavi Goel, Clement Farabet, Joelle Barral, Tris Warkentin, Omar Sanseviero, Gus Martins, Phoebe Kirk, Anand Rao, Shravya Shetty, David F. Steiner, Can Kirmizibayrak, Rory Pilgrim, Daniel Golden, and Lin Yang. MedGemma technical report, 2025. URL https://arxiv.org/abs/2507.05201.

[168] 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, 2024. URL https://arxiv.org/abs/2407.14622.

[169] 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.

[170] Glenn Shafer. Testing by betting: A strategy for statistical and scientific communication. *Journal of the Royal Statistical Society: Series A*, 184(2):407–431, 2021.

[171] Glenn Shafer, Alexander Shen, Nikolai Vereshchagin, and Vladimir Vovk. Test martingales, Bayes factors and p-values, 2011.

[172] Shubhanshu Shekhar and Aaditya Ramdas. Nonparametric two-sample testing by betting. *IEEE Transactions on Information Theory*, 70(2):1178–1203, 2023.

[173] 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`.

[174] Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis K. Titsias. Simplified and generalized masked diffusion for discrete data. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

[175] Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Yarin Gal, Nicolas Papernot, and Ross Anderson. The curse of recursion: Training on generated data makes models forget. *arXiv preprint arXiv:2305.17493*, 2023.

[176] Mukul Singh, José Cambronero, Sumit Gulwani, Vu Le, Carina Negreanu, and Gust Verbruggen. Codefusion: A pre-trained diffusion model for code generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 11697–11708, 2023.

[177] Karan Singhal, Shekoofeh Azizi, Tao Tu, S. Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, Perry Payne, Martin Seneviratne, Paul Gamble, Chris Kelly, Abubakr Babiker, Nathanael Schärli, Aakanksha Chowdhery, Philip Mansfield, Dina Demner-Fushman, Blaise Agüera y Arcas, Dale Webster, Greg S. Corrado, Yossi Matias, Katherine Chou, Juraj Gottweis, Nenad Tomasev, Yun Liu, Alvin Rajkomar, Joelle Barral, Christopher Semturs, Alan Karthikesalingam, and Vivek Natarajan. Large language models encode clinical knowledge. *Nature*, 620(7972):172–180, 2023. doi: 10.1038/s41586-023-06291-2. URL `https://doi.org/10.1038/s41586-023-06291-2`.

[178] Karan Singhal, Tao Tu, Juraj Gottweis, Rory Sayres, Ellery Wulczyn, Mohamed Amin, Le Hou, Kevin Clark, Stephen R. Pfohl, Heather Cole-Lewis, Darlene Neal, Qazi Mamunur Rashid, Mike Schaekermann, Amy Wang, Dev Dash, Jonathan H. Chen, Nigam H. Shah, Sami Lachgar, Philip Andrew Mansfield, Sushant Prakash, Bradley Green, Ewa Dominowska, Blaise Agüera y Arcas, Nenad Tomašev, Yun Liu, Renee Wong, Christopher Semturs, S. Sara Mahdavi, Joelle K. Barral, Dale R. Webster, Greg S. Corrado, Yossi Matias, Shekoofeh Azizi, Alan Karthikesalingam, and Vivek Natarajan. Toward expert-level medical question answering with large language models. *Nature Medicine*, 31(3):943–950, 2025. doi: 10.1038/s41591-024-03423-7. URL `https://doi.org/10.1038/s41591-024-03423-7`.

[179] 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.

[180] 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.

[181] 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, 2024. URL https://arxiv.org/abs/2407.10457.

[182] 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*, 2024.

[183] Volker Strassen. The existence of probability measures with given marginals. *The Annals of Mathematical Statistics*, 36(2):423–439, 1965.

[184] 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.

[185] Ruixiang Tang, Yu-Neng Chuang, and Xia Hu. The science of detecting LLM-generated texts. *arXiv preprint arXiv:2303.07205*, 2023.

[186] 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.

[187] Tooliense. Crux: Autonomous mathematical research through hierarchical multi-agent orchestration, 2025. URL https://tooliense.com/crux1.html. IC-RL Implementation with Self-Evolve Mechanism.

[188] Flemming Topsøe. Bounds for entropy and divergence for distributions over a two-element set. *J. Ineq. Pure Appl. Math*, 2(2), 2001.

[189] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[190] Ashish Venugopal, Jakob Uszkoreit, David Talbot, Franz Och, and Juri Ganitkevitch. Watermarking the outputs of structured prediction with an application in statistical machine translation. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1363–1372, Edinburgh, Scotland, UK., July 2011.

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

[192] Vladimir Vovk and Ruodu Wang. E-values: Calibration, combination and applications. *The Annals of Statistics*, 49(3):1736–1754, 2021.

[193] Vladimir G Vovk. A logic of probability, with application to the foundations of statistics. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 55(2):317–341, 1993.

[194] Abraham Wald. *Sequential Analysis*. John Wiley & Sons, New York, 1947.

[195] 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`.

[196] Lean Wang, Wenkai Yang, Deli Chen, Hao Zhou, Yankai Lin, Fandong Meng, Jie Zhou, and Xu Sun. Towards codable watermarking for injecting multi-bits information to llms. *arXiv preprint arXiv:2307.15992*, 2023.

[197] Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, Yuqiong Liu, An Yang, Andrew Zhao, Yang Yue, Shiji Song, Bowen Yu, Gao Huang, and Junyang Lin. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. *arXiv preprint arXiv:2506.01939*, 2025.

[198] 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`.

[199] Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37:95266–95290, 2024.

[200] Yue Wang, Qiuzhi Liu, Jiahao Xu, Tian Liang, Xingyu Chen, Zhiwei He, Linfeng Song, Dian Yu, Juntao Li, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. Thoughts are all over the place: On the underthinking of o1-like llms. *arXiv preprint arXiv:2501.18585*, 2025.

[201] Yuhang Wang, Feiming Xu, Zheng Lin, Guangyu He, Yuzhe Huang, Haichang Gao, Zhenxing Niu, Shiguo Lian, and Zhaoxiang Liu. From assistant to double agent: Formalizing and benchmarking attacks on OpenClaw for personalized local AI agent, 2026. URL https://arxiv.org/abs/2602.08412.

[202] Ian Waudby-Smith, Ricardo Sandoval, and Michael I Jordan. Universal log-optimality for general classes of e-processes and sequential hypothesis tests. *arXiv preprint arXiv:2504.02818*, 2025.

[203] 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, 2022.

[204] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35: 24824–24837, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html.

[205] Qingyan Wei, Yaojie Zhang, Zhiyuan Liu, Puyu Zeng, Yuxuan Wang, Biqing Qi, Dongrui Liu, and Linfeng Zhang. Accelerating diffusion large language models with slowfast sampling: The three golden principles. *arXiv preprint arXiv:2506.10848*, 2025. URL https://arxiv.org/abs/2506.10848.

[206] 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.

[207] Chengyue Wu, Hao Zhang, Shuchen Xue, Zhijian Liu, Shizhe Diao, Ligeng Zhu, Ping Luo, Song Han, and Enze Xie. Fast-dllm: Training-free acceleration of diffusion llm by enabling kv cache and parallel decoding. *arXiv preprint arXiv:2505.22618*, 2025.

[208] Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W. White, Doug Burger, and Chi Wang. AutoGen: Enabling next-gen LLM applications via multi-agent conversation, 2023. URL https://arxiv.org/abs/2308.08155.

[209] 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*, 2025. URL https://openreview.net/forum?id=VNckp7JEHn.

[210] Yuyang Wu, Yifei Wang, Ziyu Ye, Tianqi Du, Stefanie Jegelka, and Yisen Wang. When more is less: Understanding chain-of-thought length in llms, 2025. URL https://arxiv.org/abs/2502.07266.

[211] Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, Yitao Liu, Yiheng Xu, Shuyan Zhou, Silvio Savarese, Caiming Xiong, Victor Zhong, and Tao Yu. OSWorld: Benchmarking multimodal agents for open-ended tasks in real computer environments, 2024. URL https://arxiv.org/abs/2404.07972.

[212] 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.

[213] Borui Yang, Wei Li, Liyao Xiang, and Bo Li. Towards code watermarking with dual-channel transformations. *arXiv preprint arXiv:2309.00860*, 2023.

[214] Dongchao Yang, Jianwei Yu, Helin Wang, Wen Wang, Chao Weng, Yuexian Zou, and Dong Yu. Diffsound: Discrete diffusion model for text-to-sound generation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:1720–1733, 2023.

[215] John Yang, Carlos E. Jimenez, Alexander Wettig, Kilian Lieret, Shunyu Yao, Karthik Narasimhan, and Ofir Press. SWE-agent: Agent-computer interfaces enable automated software engineering, 2024. URL https://arxiv.org/abs/2405.15793.

[216] Xi Yang, Jie Zhang, Kejiang Chen, Weiming Zhang, Zehua Ma, Feng Wang, and Nenghai Yu. Tracing text provenance via context-aware lexical substitution. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 11613–11621, 2022.

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

[218] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=WE_vluYUL-X`.

[219] Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains, 2024. URL `https://arxiv.org/abs/2406.12045`.

[220] Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Dream 7b: Diffusion large language models. *arXiv preprint arXiv:2508.15487*, 2025.

[221] KiYoon Yoo, Wonhyuk Ahn, and Nojun Kwak. Advancing beyond identification: Multi-bit watermark for large language models. *arXiv preprint arXiv:2308.00221*, 2023. URL `https://arxiv.org/abs/2308.00221`.

[222] Runpeng Yu, Xinyin Ma, and Xinchao Wang. Dimple: Discrete diffusion multimodal large language model with parallel decoding. *arXiv preprint arXiv:2505.16990*, 2025.

[223] 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`.

[224] Yuxuan Song et al. Seed diffusion: A large-scale diffusion language model with high-speed inference. *arXiv preprint arXiv:2508.02193*, 2025.

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

[226] Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. Defending against neural fake news. *Advances in Neural Information Processing Systems*, 32, 2019.

[227] 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*, 2024. URL `https://openreview.net/forum?id=8rcFOqEud5`.

[228] Hanlin Zhang, Benjamin L Edelman, Danilo Francati, Daniele Venturi, Giuseppe Ateniese, and Boaz Barak. Watermarks in the sand: Impossibility of strong watermarking for generative models. *arXiv preprint arXiv:2311.04378*, 2023.

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

[230] Qiyuan Zhang, Fuyuan Lyu, Zexu Sun, Lei Wang, Weixu Zhang, Wenyue Hua, Haolun Wu, Zhihan Guo, Yufei Wang, Niklas Muennighoff, Irwin King, Xue Liu, and Chen Ma. A survey on test-time scaling in large language models: What, how, where, and how well? *arXiv preprint arXiv:2503.24235*, 2025. URL `https://arxiv.org/abs/2503.24235`.

[231] 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)*, 2024. URL `https://aclanthology.org/2024.findings-acl.924/`.

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

[233] 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, editors, *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 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/`.

[234] Xuandong Zhao, Prabhanjan Ananth, Lei Li, and Yu-Xiang Wang. Provable robust watermarking for AI-generated text. *arXiv preprint arXiv:2306.17439*, 2023.

[235] Xuandong Zhao, Lei Li, and Yu-Xiang Wang. Permute-and-flip: An optimally stable and watermarkable decoder for LLMs. *arXiv preprint arXiv:2402.05864*, 2024. URL `https://arxiv.org/abs/2402.05864`.

[236] Kaiwen Zheng, Yongxin Chen, Hanzi Mao, Ming-Yu Liu, Jun Zhu, and Qinsheng Zhang. Masked diffusion models are secretly time-agnostic masked models and exploit inaccurate categorical sampling. *arXiv preprint arXiv:2409.02908*, 2024.

[237] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. In *Advances in Neural Information Processing Systems*, volume 36, pages 46595–46623, 2023. URL `https://papers.nips.cc/paper_files/paper/2023/hash/91f18a1287b398d378ef22505bf41832-Abstract-Datasets_and_Benchmarks.html`.

[238] Lin Zheng, Jianbo Yuan, Lei Yu, and Lingpeng Kong. A reparameterized discrete diffusion model for text generation. *arXiv preprint arXiv:2302.05737*, 2023.

[239] Fengqi Zhu, Rongzhen Wang, Shen Nie, Xiaolu Zhang, Chunwei Wu, Jun Hu, Jun Zhou, Jianfei Chen, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Llada 1.5: Variance-reduced preference optimization for large language diffusion models. *arXiv preprint arXiv:2505.19223*, 2025.

Appendix A

Appendix for Sample Complexity and Expressiveness of Test-Time Scaling Paradigms

A.1 Proofs

A.1.1 Proof of Theorem 2.3.1

Proof. **Upper bound.** Let $N(o)$ denote the number of samples equal to o. For each $o \neq o^\star$, define

$$Z_t^{(o)} := \mathbf{1}\{X_t = o^\star\} - \mathbf{1}\{X_t = o\} \in [-1, 1].$$

Then

$$\mathbb{E}[Z_t^{(o)}] = p(o^\star) - p(o) \geq \Delta.$$

By Hoeffding's inequality,

$$\mathbb{P}(N(o^\star) \leq N(o)) = \mathbb{P}\left[\sum_{t=1}^{n} Z_t^{(o)} \leq 0\right] \leq \exp\left[-\frac{n\Delta^2}{2}\right].$$

Taking a union bound over all $O - 1$ incorrect answers gives

$$\mathbb{P}(\exists o \neq o^\star : N(o) \geq N(o^\star)) \leq (O-1)\exp\left[-\frac{n\Delta^2}{2}\right] \leq \delta.$$

Lower bound. Consider the two-answer instance

$$p(o^\star) = \frac{1+\Delta}{2}, \qquad p(o_2) = \frac{1-\Delta}{2}.$$

Self-consistency succeeds iff $N(o^\star) > N(o_2)$ up to tie-breaking. Let

$$S_n := \sum_{t=1}^{n} Z_t, \qquad Z_t = \begin{cases} +1, & X_t = o^\star, \\ -1, & X_t = o_2. \end{cases}$$

Then $\mathbb{E}[Z_t] = \Delta$ and $\mathrm{Var}(Z_t) = 1 - \Delta^2$. For $n \le c/\Delta^2$, choosing a sufficiently small numerical constant $c > 0$, Claim A.1.5 gives

$$\Pr(S_n \le 0) \ge c_0$$

for a universal constant $c_0 > 0$. Therefore self-consistency fails with constant probability.

□

A.1.2 Proof of Theorem 2.3.2

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

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

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

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

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

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

Lower bound. Choose $O \ge (1 - \Delta)/\Delta$ and set

$$p(1) = \Delta + \frac{1 - \Delta}{O}, \qquad p(2) = \cdots = p(O) = \frac{1 - \Delta}{O}.$$

Then $p(1) - p(2) = \Delta$ and $p(1) \le 2\Delta$. Hence, for $n \le 1/\Delta$,

$$\Pr(\text{success}) = 1 - (1 - p(1))^n \le 1 - (1 - 2\Delta)^{1/\Delta} \le 1 - e^{-4} < 0.99$$

for sufficiently small Δ.

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

A.1.3 Proof of Proposition 2.4.2

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.1.1 (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*

$\widetilde{f} = \phi(f)$ 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 $\|(\mathbf{Q}_h^{(l)})^\top \mathbf{K}_h^{(l)}\|_2 \le B_{qk}$, $\|\mathbf{V}_h^{(l)}\|_2 \le B_v$, 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.1.3, 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 $i \ge j_1, j_2, j_3$:

$$\begin{aligned}(\alpha_i + \beta_1)^\top A_0(\alpha_{j_1} + \beta_1) = (\alpha_i + \beta_1)^\top A_0(\alpha_{j_2} + \beta_1) \ge (\alpha_i + \beta_1)^\top A_0(\alpha_{j_1} + \beta_0) + C_0 \\ (\alpha_i + \beta_0)^\top A_0(\alpha_i + \beta_0) \ge (\alpha_i + \beta_0)^\top A_0(\alpha_{j_1} + \beta_1) + C_0,\end{aligned} \tag{A.1}$$

2. For any $i > j$

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

3. For any $i \ge j, j_1$

$$\begin{aligned}(\alpha_i + \beta_1)^\top A_1(\alpha_j + \beta_0) = (\alpha_i + \beta_1)^\top A_1(\alpha_{j_1} + \beta_1) + C_0 \\ (\alpha_i + \beta_1)^\top A_1(\alpha_i + \beta_1) \ge \max\{(\alpha_i + \beta_1)^\top A_1(\alpha_{j_1} + \beta_1), (\alpha_i + \beta_1)^\top A_1(\alpha_{j_1} + \beta_0)\} + C_0.\end{aligned} \tag{A.3}$$

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 $l = 1, \ldots, L$,

$$\widetilde{X}_i^{(l)} = \begin{pmatrix} X_i^{(l)} \\ \widetilde{\alpha}_i \end{pmatrix}, \ \forall i = i_1, \ldots, i_m \tag{A.4}$$

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

$$\|X_i^{(l)}\|_2 \le B_\theta (H B_v)^l, \tag{A.5}$$

and

$$\widetilde{X}_j^{(l)} = \begin{pmatrix} 0 \\ \widetilde{\alpha}_j \end{pmatrix}, \ \forall j \notin \{i_1, \ldots, i_m\} \tag{A.6}$$

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. (A.4)**.** Suppose Eq. (A.4) and Eq. (A.6) 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. (A.1) 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 $i, j_1, j_2 \in \{i_1, \ldots, i_m\}$

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

where we use the fact that $C_0 \geq C + 2\max_{h,l,i,j} \ (\mathbf{Q}_h^{(l)} X_i^{(l)})^\top (\mathbf{K}_h^{(l)} X_j^{(l)})$. Since the transformers have precision ϵ and $C \geq 2\max_{h,l,i,j} \ (\mathbf{Q}_h^{(l)} X_i^{(l)})^\top (\mathbf{K}_h^{(l)} X_j^{(l)}) \ + \log(1/\epsilon)$, 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. (A.2) 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 weights concentrate 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.(A.4) for $(l+1)$-th layer.

Prove Eq. (A.5)**.** 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. (A.22) for $l+1$.

Prove Eq. (A.6)**.** Notice that Eq. (A.1) 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 complement of $\{i : v_i \in \mathcal{V}_1\}$ itself, and therefore Eq. (A.6) follows by a simple induction argument.

Finally, at the output layer

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

This establishes the desired statement. □

Now we return to the proof of Proposition 2.4.2.

Proof. By Proposition A.1.1, 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.1.1.

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.1.3, there exists $\alpha_1, \ldots, \alpha_N, \beta_0, \beta_1, \ldots, \beta_K \in \mathbb{R}^{d_0}$ and $A, A_0, A_1, \ldots, A_K \in \mathbb{R}^{d_0 \times d_0}$ for $d_0 \leq O(K + \log N_{\max})$ such that

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

$$\begin{aligned}(\alpha_i + \beta_k)^\top A_0 (\alpha_{j_1} + \beta_{k'}) = (\alpha_i + \beta_k)^\top A_0 (\alpha_{j_2} + \beta_{k''}) \geq (\alpha_i + \beta_k)^\top A_0 (\alpha_{j_1} + \beta_0) + C_0 \\ (\alpha_i + \beta_0)^\top A_0 (\alpha_i + \beta_0) \geq (\alpha_i + \beta_0)^\top A_0 (\alpha_{j_1} + \beta_k) + C_0,\end{aligned} \tag{A.7}$$

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

$$\begin{aligned}(\alpha_i + \beta_k)^\top A (\alpha_i + \beta_k) &\geq (\alpha_i + \beta_k)^\top A (\alpha_j + \beta_{k'}) + C_0 \\ &\geq (\alpha_i + \beta_k)^\top A (\alpha_j + \beta_0) + 2C_0,\end{aligned} \tag{A.8}$$

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

$$\begin{aligned}(\alpha_i+\beta_k)^\top A_{k'}(\alpha_j+\beta_0) &\geq (\alpha_i+\beta_k)^\top A_{k'}(\alpha_{j_1}+\beta_{k''})+C_0\\ (\alpha_i+\beta_k)^\top A_k(\alpha_i+\beta_k) &\geq \max\{(\alpha_i+\beta_k)^\top A_k(\alpha_{j_1}+\beta_{k''}),(\alpha_i+\beta_k)^\top A_{k'}(\alpha_{j_1}+\beta_0)\}+C_0,\end{aligned} \tag{A.9}$$

We define ϕ as follows: for any Transformers

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

over $\mathcal{V}_k,\ k\in[K]$, the Transformer $\widetilde{f} = \phi(f_1,\ldots,f_K)$ is given by

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

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

$$\widetilde{\mathrm{pe}}\begin{pmatrix}x\\ y\end{pmatrix}; v_1,\ldots,v_i\Big) = \begin{pmatrix}\mathrm{pe}_1(x;u)\\ \vdots\\ \mathrm{pe}_K(x;u)\\ \alpha_i+y\end{pmatrix},$$

where $x\in\mathbb{R}^d$ and u is the sub-sequence of v that omits v_{i_0} (if any); for $l=1,\ldots,L$ the key, query, value matrices are given by

$$\widetilde{\mathbf{K}}^{(l)}_{(k-1)H+h} = \begin{pmatrix}0 & & & & \\ & \ddots & & & \\ & & \mathbf{K}^{(l)}_{k;h} & & \\ & & & \ddots & \\ & & & & A_0\end{pmatrix},\ \widetilde{\mathbf{Q}}^{(l)}_{(k-1)H+h} = \begin{pmatrix}0 & & & & \\ & \ddots & & & \\ & & \mathbf{Q}^{(l)}_{k;h} & & \\ & & & \ddots & \\ & & & & I\end{pmatrix},$$

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

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

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 $l = 1, \ldots, L$,

$$\widetilde{X}_i^{(l)} = \begin{pmatrix} X_{1;i}^{(l)} \\ \vdots \\ X_{K;i}^{(l)} \\ \widetilde{\alpha}_i \end{pmatrix}, \ \forall i \neq i_0 \tag{A.10}$$

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

$$\|X_{k;i}^{(l)}\|_2 \leq B_\theta (KHB_v)^l. \tag{A.11}$$

and

$$\widetilde{X}_{i_0}^{(l)} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ \widetilde{\alpha}_{i_0} \end{pmatrix} \tag{A.12}$$

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. (A.10)**.** Suppose Eq. (A.10) and Eq. (A.12) hold for $1,\dots,l-1$=th layer, and consider l-the layer. We have

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

Eq. (A.7) ensures that for any $j_1<j_2\le i$ such that $i_0\notin\{i,j_1,j_2\}$:

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

and

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

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.

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

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. (A.8) we have for any $j<i$

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

and hence the attention weighs concentrates on i itself. Thus

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

Combining these two terms, we confirm that Eq.(A.10) holds for $(l+1)$-th layer.

Prove Eq. (A.11)**.** From above,

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

This confirms Eq. (A.11) for $l+1$.

Prove Eq. (A.12)**.** Notice that Eq. (A.7) ensures that for any $j \leq i_0$:

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

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

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

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. (A.12).

Next, we show that the last layer satisfies

$$\widetilde{X}_n^{(L+1)} = \begin{pmatrix} 0 \\ \vdots \\ X_{\kappa;n}^{(L+1)} \\ \vdots \\ 0 \end{pmatrix} \tag{A.13}$$

where $X_{\kappa;n}^{(L+1)}$ is the κ-th block. To see this, we notice that Eq. (A.9) implies the followings (the proofs are identical to the above):

1. Attention sink to dummy token v_{i_0} for mismatch expert: for any $k' \neq \kappa$ and $j \leq n$ we have

$$\begin{aligned}(\widetilde{\mathbf{Q}}_{(k'-1)H+h}^{(L)} \widetilde{X}_n^{(L)})^\top (\widetilde{\mathbf{K}}_{(k'-1)H+h}^{(L)} \widetilde{X}_j^{(L)}) &= (\alpha_n + \beta_{\mathcal{E}(n)})^\top A_{k'} (\alpha_j + \beta_{\mathcal{E}(j)}) \\ &\leq (\alpha_n + \beta_{\mathcal{E}(n)})^\top A_{k'} (\alpha_{i_0} + \beta_{\mathcal{E}(i_0)}) - C \\ &= (\widetilde{\mathbf{Q}}_{(k'-1)H+h}^{(L)} \widetilde{X}_n^{(L)})^\top (\widetilde{\mathbf{K}}_{(k'-1)H+h}^{(L)} \widetilde{X}_{i_0}^{(L)}) - C. \end{aligned} \tag{A.14}$$

2. Attention to oneself for matching expert: for any $j \neq i_0$ we have

$$\begin{aligned}(\widetilde{\mathbf{Q}}_{(\kappa-1)H+h}^{(L)} \widetilde{X}_n^{(L)})^\top (\widetilde{\mathbf{K}}_{(\kappa-1)H+h}^{(L)} \widetilde{X}_j^{(L)}) &= (\alpha_n + \beta_{\mathcal{E}(n)})^\top A_{\kappa} (\alpha_j + \beta_{\mathcal{E}(j)}) \\ &\geq (\alpha_n + \beta_{\mathcal{E}(n)})^\top A_{\kappa} (\alpha_{i_0} + \beta_{\mathcal{E}(i_0)}) + C \\ &= (\widetilde{\mathbf{Q}}_{(\kappa-1)H+h}^{(L)} \widetilde{X}_n^{(L)})^\top (\widetilde{\mathbf{K}}_{(\kappa-1)H+h}^{(L)} \widetilde{X}_{i_0}^{(L)}) + C, \end{aligned} \tag{A.15}$$

and

$$\begin{aligned}(\widetilde{\mathbf{Q}}_{(\kappa-1)H+h}^{(L)} \widetilde{X}_n^{(L)})^\top (\widetilde{\mathbf{K}}_{(\kappa-1)H+h}^{(L)} \widetilde{X}_n^{(L)}) &= (\alpha_n + \beta_{\mathcal{E}(n)})^\top A_{\kappa} (\alpha_n + \beta_{\mathcal{E}(n)}) \\ &\geq (\alpha_n + \beta_{\mathcal{E}(n)})^\top A_{\kappa} (\alpha_j + \beta_{\mathcal{E}(j)}) + C \\ &= (\widetilde{\mathbf{Q}}_{(\kappa-1)H+h}^{(L)} \widetilde{X}_n^{(L)})^\top (\widetilde{\mathbf{K}}_{(\kappa-1)H+h}^{(L)} \widetilde{X}_j^{(L)}) + C. \end{aligned} \tag{A.16}$$

Combining Eq. (A.14), Eq. (A.15), and Eq. (A.16), we have

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

It follows that

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

Therefore we establish Eq. (A.13).

Finally, at the output layer

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

This establishes the desired statement. □

A.1.4 Proof of Proposition 2.4.4

Proof. Set constants B_v, B_{qk}, B_θ such that for any layer l and head h, it holds that $\|(\mathbf{Q}_h^{(l)})^\top \mathbf{K}_h^{(l)}\|_2 \le B_{qk}$, $\|\mathbf{V}_h^{(l)}\|_2 \le B_v$, and $\|\theta(v)\|_2 \le B_\theta$ 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 $\xi_u \le i < \xi_{u+1}$ ($\xi_0 = -1, \xi_{m+1} = \infty$ by default). Let $\mathcal{E}(\cdot)$ denote the task id indicated by the special token. By Lemma A.1.2, there exists $\alpha_1,\ldots,\alpha_N,\beta_1,\ldots,\beta_K \in \mathbb{R}^{d_0}$ and $A, A_1,\ldots,A_K \in \mathbb{R}^{d_0\times d_0}$ for $d_0 \le O(K + \log N_{\max})$ such that for any $n \le N$ we have

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

$$\alpha_n^\top A_k(\alpha_n + \beta_{k'}) \ge C_0 + \begin{cases} \alpha_n^\top A_k \alpha_n \\ \alpha_n^\top A_k \alpha_j \\ \alpha_n^\top A_k(\alpha_j + \beta_{k''}) \end{cases}, \ \forall 0 \le j < n, 1 \le k'' \le K. \tag{A.17}$$

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

$$\alpha_n^\top A_k \alpha_n = \alpha_n^\top A_k \alpha_0 \ge C_0 + \begin{cases} \alpha_n^\top A_k(\alpha_n + \beta_k) \\ \alpha_n^\top A_k \alpha_j \\ \alpha_n^\top A_k(\alpha_j + \beta_{k'}) \end{cases}, \ \forall 0 < j < n, k' \ne k. \tag{A.18}$$

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

$$(\alpha_n + \beta_{k'})^\top A_k(\alpha_n + \beta_{k'}) \ge C_0 + (\alpha_n + \beta_{k'})^\top A_k \alpha_j, \ \forall 0 \le j \le n. \tag{A.19}$$

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

$$\begin{aligned} \alpha_n^\top A \alpha_n &\ge \alpha_n^\top A(\alpha_n + \beta_k) + C_0 \\ &\ge C_0 + \max\{\alpha_n^\top A \alpha_j, \alpha_n^\top A(\alpha_j + \beta_{k'})\}, \ \forall k, k'' \in [K]. \end{aligned} \tag{A.20}$$

We define ϕ as follows: for any Transformers

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

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

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

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

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

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

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

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:

$$\widetilde{X}_i^{(l)} = \begin{pmatrix} X_{1;i}^{(l)} \\ \vdots \\ X_{K;i}^{(l)} \\ \widetilde{\alpha}_i \end{pmatrix}, \ i = 1, \ldots, n, \tag{A.21}$$

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

$$\|X_{k;i}^{(l)}\|_2 \leq B_\theta (KHB_v)^l. \tag{A.22}$$

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

$$X_{k;\xi_i}^{(l)} = 0, \ \forall k = 1, \ldots, K, \tag{A.23}$$

and for $j = 1, \ldots, \xi_1 - 1$ we have

$$X_{k;j}^{(l)} = Y_{k;j}^{(l)}, \ \forall k = 1, \ldots, K, \tag{A.24}$$

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

$$X_{\kappa;j}^{(l)} = Y_{\kappa, j-\xi_m-1+\xi_1}^{(l)}, \ X_{k';j}^{(l)} = 0, \ \forall k' \neq \kappa, \tag{A.25}$$

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

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 $1, \ldots, l$, consider layer $l + 1$. We have

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

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 $i = 1, \ldots, \xi_1 - 1, \xi_m + 1, \ldots, n$, where $\zeta(i) := \begin{cases} i, & i < \xi_1 \\ i - \xi_m - 1 + \xi_1, & i > \xi_m \end{cases}$.

Notice that for $j \le i$:

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

Prove Eq (A.21)**.** By properties of α, β, A, for any $j_2 < \xi_u < j_1 < i < \xi_{u+1}$ notice that:

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

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 $\xi_u < i < \xi_{u+1}$

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

and for $i = \xi_u$

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

where

$$X_{k;i}^{(l+1)} = \sum_{k=1}^{K}\sum_{h=1}^{H}\sum_{j=1}^{i} \frac{\exp\left((\widetilde{\mathbf{Q}}_{(k-1)H+h}^{(l)}\widetilde{X}_i^{(l)})^\top(\widetilde{\mathbf{K}}_{(k-1)H+h}^{(l)}\widetilde{X}_j^{(l)})\right)}{\widetilde{Z}_{(k-1)H+h}^{(l)}} \cdot \mathbf{V}_{k;h}^{(l)} X_{k;j}^{(l)}. \tag{A.26}$$

This confirms Eq. (A.21) for $l+1$.

Prove Eq. (A.22)**.** From above,

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

This confirms Eq. (A.22) for $l+1$.

Prove Eq. (A.23)**.** We first show $X_{k;\xi_1}^{(l)} = 0$. Indeed, by the properties of α_t, β_k, for any $j \le \xi_1$

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

It follows from Eq. (A.26) 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 $i' < i$,

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

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

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

where we apply the induction hypothesis $k; X^{(l)}_{\xi_j} = 0$ for all $j = 1, \ldots, i-1$. This thus completes the proof of Eq. (A.23).

Prove Eq. (A.24)**.** When $j_1 < j_2 \le i < \xi_1$, we have

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

It follows that

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

and

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

This confirms Eq. (A.24).

Prove Eq. (A.25)**.** 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

$$(\widetilde{\mathbf{Q}}^{(l)}_{(k'-1)H+h}\widetilde{X}^{(l)}_{i})^\top(\widetilde{\mathbf{K}}^{(l)}_{(k'-1)H+h}\widetilde{X}^{(l)}_{j}) \le (\widetilde{\mathbf{Q}}^{(l)}_{(k'-1)H+h}\widetilde{X}^{(l)}_{i})^\top(\widetilde{\mathbf{K}}^{(l)}_{(k'-1)H+h}\widetilde{X}^{(l)}_{\xi_m}) - C. \quad (A.27)$$

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

$$(\widetilde{\mathbf{Q}}^{(l)}_{(\kappa-1)H+h}\widetilde{X}^{(l)}_i)^\top(\widetilde{\mathbf{K}}^{(l)}_{(\kappa-1)H+h}\widetilde{X}^{(l)}_j) \ge (\widetilde{\mathbf{Q}}^{(l)}_{(\kappa-1)H+h}\widetilde{X}^{(l)}_i)^\top(\widetilde{\mathbf{K}}^{(l)}_{(\kappa-1)H+h}\widetilde{X}^{(l)}_{j'}) + C. \tag{A.28}$$

and for $j_1 < j_2 \in \{1, \ldots, \xi - 1 - 1, \xi_m + 1, \ldots, n\}$

$$\begin{aligned}&(\widetilde{\mathbf{Q}}^{(l)}_{(\kappa-1)H+h}\widetilde{X}^{(l)}_i)^\top(\widetilde{\mathbf{K}}^{(l)}_{(\kappa-1)H+h}\widetilde{X}^{(l)}_{j_1}) - (\widetilde{\mathbf{Q}}^{(l)}_{(\kappa-1)H+h}\widetilde{X}^{(l)}_i)^\top(\widetilde{\mathbf{K}}^{(l)}_{(\kappa-1)H+h}\widetilde{X}^{(l)}_{j_2})\\ =\ &(\mathbf{Q}^{(l)}_{\kappa;h}Y^{(l)}_{\kappa;i-\xi_m-1+\xi_1})^\top(\mathbf{K}^{(l)}_{\kappa;h}Y^{(l)}_{\zeta(j_1)}) - (\mathbf{Q}^{(l)}_{\kappa;h}Y^{(l)}_{i-\xi_m-1+\xi_1})^\top\mathbf{K}^{(l)}_{\kappa;h}Y^{(l)}_{\kappa;\zeta(j_2)}),\end{aligned} \tag{A.29}$$

To see Eq. (A.27), we notice that

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

where we use Eq. (A.17) with $k' \ne \kappa$.

To see Eq. (A.28), we notice that

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

and

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

where we use Eq. (A.18) and Eq. (A.20).

When $\xi_m < j_1 < j_2$, Eq. (A.29) follows directly from

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

The other cases follow similarly due to Eq. (A.20).

We have hence confirmed Eq. (A.27), Eq. (A.28), Eq. (A.29), 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

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

Therefore we establish Eq. (A.25). This completes the induction.

At the output layer, we have

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

This establishes the desired Eq. (2.2). □

A.1.5 Proof of Theorem 2.4.7

Proof. Let ϕ_s, ϕ_m, ϕ_e denote the general-purpose Transformers in Proposition 2.4.4 (with K experts), Proposition 2.4.2 (with $K=3$ token spaces), and Proposition A.1.1 (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

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

achieves the desired property.

Indeed, let $g_1 = \phi_s(\phi_e(f_1),\ldots,\phi_e(f_K))$, by Proposition 2.4.4, we have

1. **Expert following**: At t-th iteration,

$$p_{g_1}\left(\cdot \text{ prompt}\right) \sim p_{f_{a^{(t)}}}\left(\cdot\, q|u^{(t)}_{1:i-1}\right),$$

where $q|u^{(t)}_{1:i-1}$ is the token sequence obtained by concatenating the user query q and prior generated part in response t: $u^{(t)}_{1:i-1}$.

2. **Regret minimization**:

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

Therefore by Proposition 2.4.2, we have

$$u^{(t)}_i \sim p_{f_{a^{(t)}}}\left(\cdot\, q|u^{(t)}_{1:i-1}\right).$$

It follows that

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

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.1.6 Attention sink positional encoding

In this section, we introduce positional encoding mechanisms that induce attention sink behaviors used by Theorem 2.4.7.

Lemma A.1.2 (Attention Sink Positional Encoding, Type 1)**.** *For any $C \in \mathbb{R}_+$, $K, N \in \mathbb{Z}_+$, there exist vectors $\alpha_0, \alpha_1, \ldots, \alpha_N, \beta_1, \ldots, \beta_K \in \mathbb{R}^d$ and matrices $A, A_1, \ldots, A_K \in \mathbb{R}^{d\times d}$ for $d = O(K + \log N)$ such that for any $n \in [N]$ the followings hold:*

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

$$\alpha_n^\top A_k(\alpha_n + \beta_{k'}) \ge C + \max\left\{\alpha_n^\top A_k\alpha_n,\ \max_{0\le j<n} \alpha_n^\top A_k\alpha_j,\ \max_{\substack{1\le j<n\\ k''\in[K]}} \alpha_n^\top A_k(\alpha_j + \beta_{k''})\right\}.$$

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

$$\alpha_n^\top A_k\alpha_n = \alpha_n^\top A_k\alpha_0 \ge C + \max\left\{\alpha_n^\top A_k(\alpha_n + \beta_k),\ \max_{0< j<n} \alpha_n^\top A_k\alpha_j,\ \max_{\substack{0<j<n\\ k'\in[K]}} \alpha_n^\top A_k(\alpha_j + \beta_{k'})\right\}.$$

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

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

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

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

Proof. By Claim A.1.4, we can find $\gamma_1, \ldots, \gamma_N \in \mathbb{R}^{\bar{d}}$ such that $\bar{d} = O(\log N)$,

$$|\gamma_i^\top \gamma_j| \leq \frac{1}{2}, \qquad \forall i \neq j \in [N], \qquad \text{and} \qquad \gamma_i^\top \gamma_i = 1, \qquad \forall i \in [N].$$

Let $e_1, \ldots, e_K$ be the standard basis of $\mathbb{R}^K$, and let $\mathbf{1}_K$ denote the all-one vector in $\mathbb{R}^K$. We use the block decomposition

$$\mathbb{R}^d = \mathbb{R}^{\bar{d}} \oplus \mathbb{R} \oplus \mathbb{R}^K \oplus \mathbb{R}.$$

Define

$$\alpha_i = \begin{pmatrix} \gamma_i \\ 1 \\ 0 \\ 0 \end{pmatrix}, \qquad \alpha_0 = \begin{pmatrix} 0 \\ 2 \\ 0 \\ 0 \end{pmatrix}, \qquad \beta_k = \begin{pmatrix} 0 \\ 0 \\ e_k \\ 1 \end{pmatrix}.$$

For each $k \in [K]$, define $w_k \in \mathbb{R}^K$ by

$$(w_k)_{k'} = \begin{cases} -C, & k' = k, \\ C, & k' \neq k. \end{cases}$$

Let

$$A_k = \begin{pmatrix} 4C \cdot I_{\bar{d}} & 0 & 0 & 0 \\ 0 & 4C & w_k^\top & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2C \end{pmatrix}, \qquad A = \begin{pmatrix} 4C \cdot I_{\bar{d}} & 0 & 0 & 0 \\ 0 & 0 & -C \cdot \mathbf{1}_K^\top & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}.$$

The dimension can be bounded by $d = \bar{d} + K + 2 = O(K + \log N)$.

We first verify the properties involving A_k. Direct computation gives

$$\alpha_n^\top A_k \alpha_n = \alpha_n^\top A_k \alpha_0 = 8C.$$

Moreover, for $1 \leq j < n$,

$$\alpha_n^\top A_k \alpha_j = 4C\gamma_n^\top \gamma_j + 4C \leq 6C.$$

For any $k' \in [K]$,

$$\alpha_n^\top A_k(\alpha_n + \beta_{k'}) = 8C + (w_k)_{k'} = \begin{cases} 7C, & k' = k, \\ 9C, & k' \neq k. \end{cases}$$

For $1 \leq j < n$ and $k' \in [K]$,

$$\alpha_n^\top A_k(\alpha_j + \beta_{k'}) = 4C\gamma_n^\top \gamma_j + 4C + (w_k)_{k'} \leq 7C.$$

Therefore, if $k' \neq k$, then

$$\alpha_n^\top A_k(\alpha_n + \beta_{k'}) = 9C$$

is at least C larger than all terms in item 1. This proves item 1. Similarly, $\alpha_n^\top A_k \alpha_n = 8C$ is at least C larger than $\alpha_n^\top A_k(\alpha_n + \beta_k) = 7C$, all earlier unmarked scores, and all earlier marked scores. This proves item 2.

For item 3, direct computation gives

$$(\alpha_n + \beta_{k'})^\top A_k(\alpha_n + \beta_{k'}) = 10C + (w_k)_{k'} \geq 9C.$$

On the other hand, for every $0 \leq j \leq n$,

$$(\alpha_n + \beta_{k'})^\top A_k \alpha_j \leq 8C.$$

This proves item 3.

It remains to verify item 4. We have

$$\alpha_n^\top A \alpha_n = 4C, \qquad \alpha_n^\top A(\alpha_n + \beta_k) = 3C.$$

For $0 < j < n$,

$$\alpha_n^\top A \alpha_j = 4C\gamma_n^\top \gamma_j \leq 2C,$$

and

$$\alpha_n^\top A(\alpha_j + \beta_{k'}) = 4C\gamma_n^\top \gamma_j - C \leq C.$$

Thus

$$4C \geq 3C + C, \qquad 3C \geq C + \max\{2C, C\}.$$

This proves item 4 and completes the proof. □

Lemma A.1.3 (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 $A, A_0, A_1, \ldots, A_K \in \mathbb{R}^{d \times d}$ for $d = O(K + \log N)$ such that for any $n \in [N]$ the followings hold:*

1. *For any $i \geq j_1, j_2$ and $k, k', k'' \in [K]$:*

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

Moreover, for any $i \geq j_1$ and $k \in [K]$:

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

2. For any $i > j$ and $k, k' \in [K]$:

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

3. For any $i \geq j, j_1,\ k, k', k'' \in [K]$ with $k' \neq k$:

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

Moreover, for any $i \geq j_1,\ k, k', k'' \in [K]$ with $k' \neq k$ and $(j_1, k'') \neq (i, k)$:

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

Proof. By Claim A.1.4, we can find $\gamma_1, \ldots, \gamma_N \in \mathbb{R}^{\bar{d}}$ such that $\bar{d} = O(\log N)$,

$$|\gamma_i^\top \gamma_j| \leq \frac{1}{2}, \qquad \forall i \neq j \in [N], \qquad \text{and} \qquad \gamma_i^\top \gamma_i = 1, \qquad \forall i \in [N].$$

Let $e_1, \ldots, e_K$ be the standard basis of $\mathbb{R}^K$, and let $\mathbf{1}_K$ denote the all-one vector in $\mathbb{R}^K$. We use the block decomposition

$$\mathbb{R}^d = \mathbb{R}^{\bar{d}} \oplus \mathbb{R}^K \oplus \mathbb{R} \oplus \mathbb{R}.$$

Define

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

Define

$$A_0 = \begin{pmatrix}0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 2C & 0 \\ 0 & 0 & 0 & C\end{pmatrix}, \qquad A = \begin{pmatrix}2C \cdot I_{\bar{d}} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 2C & 0 \\ 0 & 0 & 0 & 0\end{pmatrix}.$$

For each $k \in [K]$, let $u_k = \mathbf{1}_K - e_k$ and define

$$A_k = \begin{pmatrix}2C \cdot I_{\bar{d}} & 0 & 0 & 0 \\ 0 & 6C \cdot e_k e_k^\top & 0 & 5C \cdot u_k \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0\end{pmatrix}.$$

The dimension can be bounded by $d = \bar{d} + K + 2 = O(K + \log N)$.

We first verify item 1. For any i, j and $k, k' \in [K]$,

$$(\alpha_i + \beta_k)^\top A_0(\alpha_j + \beta_{k'}) = 2C,$$

whereas

$$(\alpha_i + \beta_k)^\top A_0(\alpha_j + \beta_0) = 0.$$

Also,

$$(\alpha_i + \beta_0)^\top A_0(\alpha_i + \beta_0) = C, \qquad (\alpha_i + \beta_0)^\top A_0(\alpha_j + \beta_k) = 0.$$

This proves item 1.

For item 2, if $i > j$, then

$$(\alpha_i + \beta_k)^\top A(\alpha_i + \beta_k) = 4C.$$

Moreover,

$$(\alpha_i + \beta_k)^\top A(\alpha_j + \beta_{k'}) = 2C\gamma_i^\top \gamma_j + 2C \le 3C,$$

and

$$(\alpha_i + \beta_k)^\top A(\alpha_j + \beta_0) = 2C\gamma_i^\top \gamma_j \le C.$$

Therefore,

$$4C \ge 3C + C, \qquad 3C \ge C + 2C,$$

which proves item 2.

It remains to verify item 3. Direct computation gives

$$(\alpha_i + \beta_k)^\top A_{k'}(\alpha_j + \beta_{k''}) = 2C\gamma_i^\top \gamma_j + 6C \cdot \mathbf{1}\{k = k'\}\mathbf{1}\{k'' = k'\},$$

and

$$(\alpha_i + \beta_k)^\top A_{k'}(\alpha_j + \beta_0) = 2C\gamma_i^\top \gamma_j + 5C \cdot \mathbf{1}\{k \ne k'\}.$$

If $k' \ne k$, then

$$(\alpha_i + \beta_k)^\top A_{k'}(\alpha_j + \beta_0) \ge -C + 5C = 4C,$$

while

$$(\alpha_i + \beta_k)^\top A_{k'}(\alpha_{j_1} + \beta_{k''}) \le C.$$

Thus the first inequality in item 3 holds.

Finally,

$$(\alpha_i + \beta_k)^\top A_k(\alpha_i + \beta_k) = 2C + 6C = 8C.$$

For $(j_1, k'') \ne (i, k)$, either $j_1 \ne i$, in which case

$$(\alpha_i + \beta_k)^\top A_k(\alpha_{j_1} + \beta_{k''}) \le C + 6C = 7C,$$

or $j_1 = i$ and $k'' \neq k$, in which case

$$(\alpha_i + \beta_k)^\top A_k(\alpha_i + \beta_{k''}) = 2C.$$

In both cases,

$$(\alpha_i + \beta_k)^\top A_k(\alpha_{j_1} + \beta_{k''}) \leq 7C.$$

Moreover, for $k' \neq k$,

$$(\alpha_i + \beta_k)^\top A_{k'}(\alpha_{j_1} + \beta_0) \leq 2C + 5C = 7C.$$

Since the matching self-score is $8C$, the second inequality in item 3 follows. This completes the proof. □

A.1.7 Technical Claims

Claim A.1.4 (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.1.5 (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,

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

A.2 Experiment Details

A.2.1 Implementation details of self-correction experiments

The model configurations are detailed in Table A.1. Our code is implemented based on `PyTorch` [5] and `minGPT`[1]. All the models are trained on one NVIDIA GeForce RTX 2080 Ti GPU with 11GB memory.

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 dropout or weight decay during training. We repeat the experiments three times under different random seeds and report the average accuracy with error bars.

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 A.1: Model configuration hyperparameters of self-correction experiments.

A.2.2 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: `$\boxed{ANSWER}$`.'

A.3 Best-of-n with Imperfect Verification

We additionally perform experiments to test robustness of best-of-n under imperfect verification. Using Qwen3-32B as an LLM verifier, we evaluated best-of-n on the AIME 24/25 datasets with 3 recent models. As shown in Table A.2, best-of-n with an LLM verifier closely

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

matches the performance using an oracle verifier. This evidence supports the practical validity of our theoretical assumptions in Section 2.5.2.

Model	**Best-of-n (64, oracle)**	**Best-of-n (64, LLM verifier)**
Qwen3-1.7B	80.00	76.56
Qwen3-4B	91.67	85.20
Llama-3.2-3B-Instruct	23.33	21.45

Table A.2: Comparison of best-of-64 performance using an oracle verifier vs. Qwen3-32B as an LLM verifier on AIME 24/25. Results are similar across all evaluated models, indicating robustness to imperfect verification.

Appendix B

Appendix for Bits-to-Rounds Sampling for Diffusion Language Model Inference

B.1 Proof of Theorem 3.3.2

Proof. We first control the approximation error incurred by factorizing the joint likelihood into per-group conditionals. For any ordered partition $(A_1, \ldots, A_R)$ as above, by the ordered-partition chain rule of probability density function, we have

$$-\log p(\mathbf{x}) \;=\; \sum_{r=1}^{R}\Big[-\log p(x^{A_r} \mid x^{C_r})\Big].$$

For each round $r \in [R]$, by Assumption 3.3.1 and Fréchet inequality, the joint conditional probability of x^{A_r} given the past unmasked set C_r can be lower bounded by

$$p(x^{A_r} \mid x^{C_r}) \geq 1 - \sum_{i \in A_r} (1 - p(x^i | x^{C_r})) \geq 1 - \frac{f|A_r|}{1 + |A_r|}. \tag{B.1}$$

Thus, the entire negative joint log probability $-\log p(\mathbf{x})$ can be bounded by

$$\begin{aligned}
-\log p(\mathbf{x}) &= \sum_{r=1}^{R}\Big[-\log p(x^{A_r} \mid x^{C_r})\Big] \\
&\leq \sum_{r=1}^{R} \log \left(\frac{1}{1 - f|A_r|/(1 + |A_r|)} \right) && \text{(plugging in (B.1))} \\
&= \sum_{r=1}^{R} \log \left(\frac{1 + |A_r|}{1 + (1 - f)|A_r|} \right) \\
&\leq R \log \left(\frac{1 + n}{1 + (1 - f)n} \right). && \left(\tfrac{1+t}{1+(1-f)t} \text{ is monotone increasing with } t > 0\right)
\end{aligned}$$

Rearranging, we obtain

$$R \geq \frac{-\log p(\mathbf{x})}{\log\left(\frac{1+n}{1+(1-f)n}\right)}. \tag{B.2}$$

On the other hand, by the definition of ϵ in (3.1), we have

$$\sum_{r=1}^{R} \log p(x^{A_r} \mid x^{C_r}) - \sum_{r=1}^{R} \sum_{i \in A_r} \log p(x^i \mid x^{C_r}) \ \leq \epsilon \tag{B.3}$$

for any $r \in [R]$. Consequently, again by applying the chain rule and summing up over r,

$$\begin{aligned}
-\log p(\mathbf{x}) &= \sum_{r=1}^{R} \left[-\log p(x^{A_r} \mid x^{C_r}) - \sum_{i \in A_r} \Big[-\log p(x^i \mid x^{C_r}) \Big] + \sum_{i \in A_r} \Big[-\log p(x^i \mid x^{C_r}) \Big] \right] \\
&\leq \epsilon + \sum_{r=1}^{R} \sum_{i \in A_r} \Big[-\log p(x^i \mid x^{C_r}) \Big] && \text{(plugging in (B.3))} \\
&\leq \epsilon + \sum_{r=1}^{R} \left[-\sum_{i \in A_r} \log \quad 1 - \frac{f}{1+|A_r|} \right) \Bigg] && \text{(by Assumption 3.3.1)} \\
&= \epsilon + \sum_{r=1}^{R} \left[|A_r| \log \quad 1 + \frac{f}{(1-f)+|A_r|} \right) \Bigg] \\
&\leq \epsilon + \sum_{r=1}^{R} \frac{|A_r| f}{1-f+|A_r|} && (\log(1+t) \leq t \text{ for } t \geq 0.) \\
&\leq \epsilon + Rf.
\end{aligned}$$

Rearranging, we obtain

$$R \geq \frac{-\log p(\mathbf{x}) - \epsilon}{f}. \tag{B.4}$$

Combining Eq. (B.2) and Eq. (B.4), we have

$$R \geq \max\left\{ \frac{-\log p(\mathbf{x}) - \epsilon}{f}, \frac{-\log p(\mathbf{x})}{\log\left(\frac{1+n}{1+(1-f)n}\right)} \right\}.$$

The proof is complete. □

B.2 Additional Figures, Examples, and Experiments

B.2.1 Examples of the inefficiency of confidence-based parallel decoding algorithm

Example 1: Structured Profile Records Consider randomly sampling the profile of an undergraduate student from a structured database:

$$\mathbf{x} = [\texttt{name}, \texttt{age}, \texttt{school}, \texttt{hobby}] \sim p_{\text{data}}.$$

A confidence-based decoding algorithm typically selects the field `age` first, since its predicted distribution is concentrated within a narrow range (18–22), yielding artificially high confidence. However, if one decodes `name` first—despite its relatively diffuse predictive distribution—the remaining fields often become nearly deterministic: the student's `school`, `age`, and `hobby` are strongly constrained once the identity is known. Thus, a naive "confident-token-first" decoding strategy is locally optimal but globally inefficient: it picks the token that resolves the *least* uncertainty.

Example 2: Code Generation A similar phenomenon arises in coding tasks. Consider predicting the following masked snippet for computing a factorial:

```
def factorial(n):
    [MASK] = 1
    for i in range(1, n + 1):
        [MASK] *= i
    return [MASK]
```

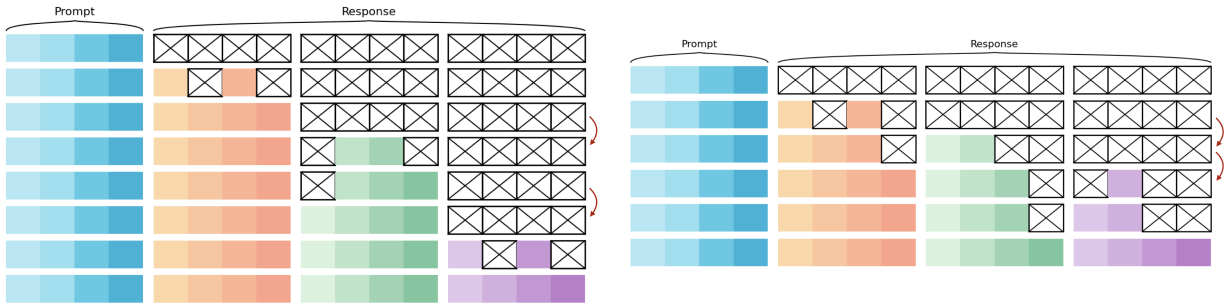
Each possible masked variable name (e.g., `ans`, `result`, `res`) may have low marginal conditional probability due to the arbitrary naming convention, resulting in low-confidence among all three masked tokens. Nevertheless, decoding *any one* of the masked slots immediately forces the remaining masks to adopt the same name, thereby collapsing the search space for all subsequent predictions. Again, high-entropy tokens, if decoded strategically, can unlock substantial determinism elsewhere.

B.2.2 Visualization of block sampling LLaDA and fast block diffusion sampling

Figure B.1 provides a visualization of both block sampling LLaDA [139] (Panel (a)) and our fast block diffusion sampling approach (Panel (b)) for comparison.

B.2.3 Additional experiments for fairer comparison

We report an additional performance curve of ETE called ETE (Standard). We measure performance across a range of confidence thresholds while holding other hyperparameters



(a) Block diffusion (LLaDA) sampling [139]. (b) Ours: Fast block diffusion sampling.

Figure B.1: Panel (a): Block diffusion unlocks the next block after the current block is fully unmasked; Panel (b): Fast block diffusion (with budget 1) unlocks the next block after the budget exhausts and retains the ability to unmask prior blocks, enabling faster decoding. In both figures, we use red arrows to indicate unlocking the next block.

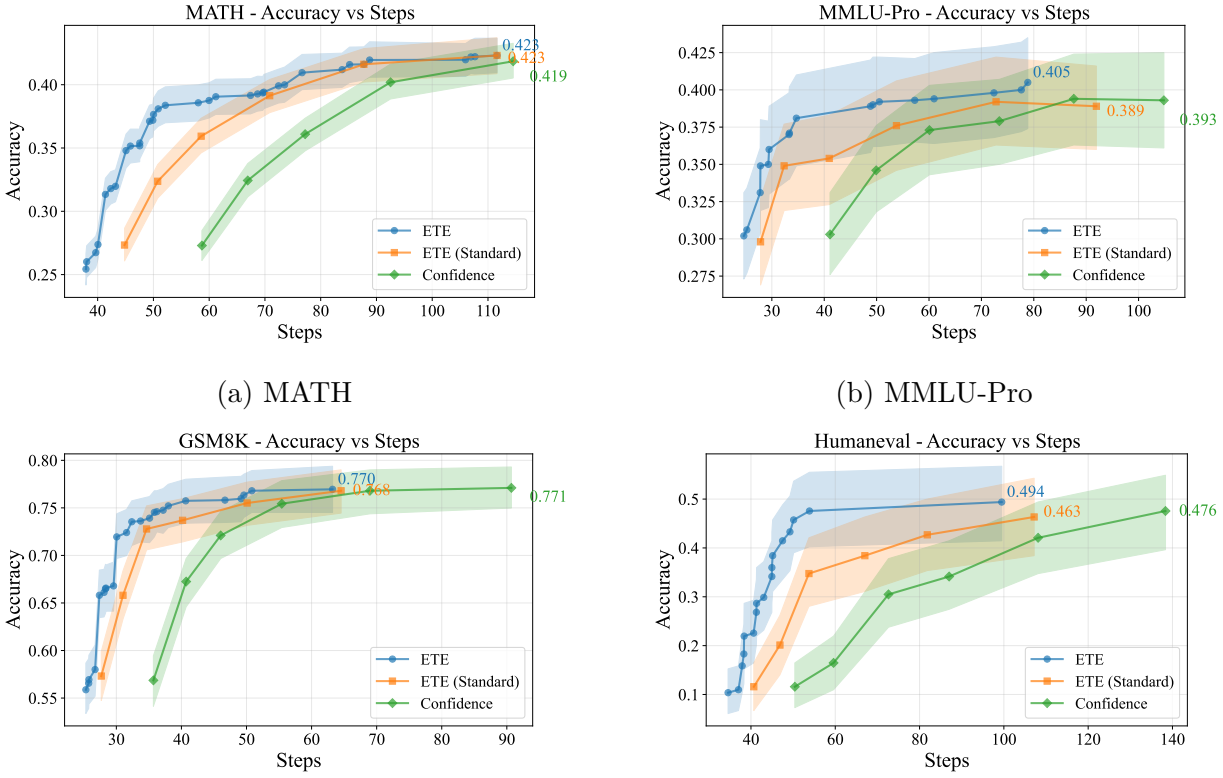
fixed (tuned on the GSM8K training set) so that both ETE (Standard) and confidence-based baseline only have one degree of freedom (confidence thresholds). This ensures a direct and fair comparison with the baseline by varying the same primary hyperparameter (the static confidence threshold). We still keep the original ETE curve in Figure 3.6 for comparison.

According to Figure B.2, in direct comparisons using fixed hyperparameters, the ETE (Standard) frontier is still consistently superior to that of the confidence-based baseline. ETE (Standard) attains higher average accuracy at comparable or lower step budgets, highlighting statistically significant improvements particularly in low-compute regimes. Moreover, by leveraging the additional degree of freedom in step scheduling, the ETE frontier achieves an even more favorable tradeoff, reaching higher peak accuracies with fewer steps. Overall, these results demonstrate that ETE allocates computational resources more effectively, translating to superior accuracy-per-step efficiency across diverse domains.

B.3 Confidence-based Parallel Decoding Algorithms and Detailed Illustrations of ETE

B.3.1 Confidence-based parallel decoding algorithms

Algorithm B.3.1 demonstrates the standard confidence-based parallel decoding process with a static confidence threshold C.



(a) MATH

(b) MMLU-Pro

(c) GSM8K

(d) HumanEval

Figure B.2: Accuracy-steps frontiers of ETE with an additional ETE (Standard) curve versus baseline on four benchmarks.

Algorithm B.3.1 Confidence-based parallel decoding with block diffusion sampling

Input: Model p_θ, prompt $\mathbf{x}_{\text{prompt}}$, generation length n, num blocks L, confidence threshold C.
$\mathbf{x}_0 \leftarrow \left(\mathbf{x}_{\text{prompt}}, [\mathrm{M}]^{\otimes n}\right),\ t \leftarrow 0$
$n_0 = |\mathbf{x}_{\text{prompt}}|,\ \mathcal{B}_b = \left[n_0 + \frac{(b-1)n}{L} : n_0 + \frac{bn}{L}\right]\ (b = 1, \ldots, L)$
for $b = 1 \to L$ **do**
  while block b is not fully unmasked **do**
    $S \leftarrow \{0 \le i \le n_0 + n : x_t^i = [\mathrm{M}]\} \cap \mathcal{B}_b$
    $\hat{x}_{t+1}^i \leftarrow \arg\max_{v \in \mathcal{V}} p_\theta^i(v \mid \mathbf{x}_t),\ i \in S$
    $x_{t+1}^i \leftarrow \begin{cases} \hat{x}_{t+1}^i, & \text{if } p_\theta^i(\hat{x}_{t+1}^i \mid \mathbf{x}_t) > C, \\ x_t^i, & \text{otherwise.} \end{cases}$
    if $\mathbf{x}_{t+1} = \mathbf{x}_t$ **then**
      $i^*, \hat{x}_{t+1}^{i^*} \leftarrow \arg\max_{i \in S, v \in \mathcal{V}} p_\theta^i(v \mid \mathbf{x}_t),\ x_{t+1}^{i^*} \leftarrow \hat{x}_{t+1}^{i^*}$
    $t \leftarrow t + 1$
Return $\mathbf{x}_t$

B.3.2 Detailed illustration of ETE

We provide the detailed illustration of ETE in Algorithm B.3.2, which orchestrates exploitation and exploration phases within the fast block diffusion framework. Algorithms B.3.3-B.3.6 elaborate on the detailed subroutines of exploitation, implicit exploration, the trigger for targeted exploration and targeted exploration, respectively.

Algorithm B.3.2 ETE Main Sampling Procedure

Input: Model p, prompt $\mathbf{x}_{\text{prompt}}$, sequence length n, num blocks L, steps per block N, confidence threshold C, additional decoding rounds n_f, exploration params $c^{\text{info}}, \gamma, \alpha, \beta, N_e$.
$\mathbf{x}_0 \leftarrow \left(\mathbf{x}_{\text{prompt}}, [\text{M}]^{\otimes n}\right),\ t \leftarrow 0$
for $b = 1 \to L$ **do**
  $t^b \leftarrow t$
  while $t - t^b \leq N$ and $\mathcal{M}_t \neq \emptyset$ **do**
    $\mathbf{x}_{t+1} \leftarrow \textsc{Exploit}(\mathbf{x}_t, b, C)$
    if $\textsc{TriggerExplore}(\mathbf{x}_{t+1}, b, \gamma, N_e)$ **then**
      $\mathbf{x}_{t+1} \leftarrow \textsc{ImplicitExplore}(\mathbf{x}_{t+1}, b-1)$
      $\mathbf{x}_{t+2} \leftarrow \textsc{TargetedExplore}(\mathbf{x}_{t+1}, b, \alpha, \beta, c^{\text{info}})$
      $t \leftarrow t+2$
    else
      $\mathbf{x}_{t+1} \leftarrow \textsc{ImplicitExplore}(\mathbf{x}_{t+1}, b)$
      $t \leftarrow t+1$
  if $\mathcal{M}_t = \emptyset$ **then**
    break
if $\mathcal{M}_t \neq \emptyset$ **then**
  for $\_ = 1 \to n_f$ **do**
    $\mathbf{x}_{t+1} \leftarrow \textsc{Exploit}(\mathbf{x}_t, L, C)$
    $t \leftarrow t+1$
Return $\mathbf{x}_t$

Algorithm B.3.3 EXPLOIT Subroutine

Input: $\mathbf{x}_t, b, C$
Form the feasible set S_t as in Eq. (3.4).
Compute exploitation tokens $\mathbf{x}_{t+1}$ according to Eq. (3.5).
return $\mathbf{x}_{t+1}$

Algorithm B.3.4 IMPLICITEXPLORE Subroutine

Input: $\mathbf{x}_{t+1}, b$
Reuse $c(\hat{x}^i_{t+1})$ from the last forward pass.
for $b' = 1, \ldots, b$ **do**
    $S_{b';t} \leftarrow S_t \cap \mathcal{B}_{b'}$ (Parallel implicit exploration across active blocks)
    if $S_{b';t} = \emptyset$ **then**
        continue
    Let $U_{b';t}$ be the subset of $S_{b';t}$ not already committed by Eq. (3.5).
    if $U_{b';t} \neq \emptyset$ **then**
        $i^\star \leftarrow \arg\max_{i \in U_{b';t}} p^i_\theta(\hat{x}^i_{t+1} \mid \mathbf{x}_t)$
        $x^{i^\star}_{t+1} \leftarrow \hat{x}^{i^\star}_{t+1}$ (One highest-confidence token per block)
return $\mathbf{x}_{t+1}$

Algorithm B.3.5 TRIGGEREXPLORE Subroutine

Input: $\mathbf{x}_{t+1}, b, \gamma, N_e$
$i_{\text{frontier}} = \min(bn_b, \max((b-1)n_b, \arg\max_i\{i : x^i_t \neq [\text{M}]\}) + n_b/2)$
$\mathcal{W}_t \leftarrow \{i : x^i_t = [\text{M}], i \in ((b-1)n_b, i_{\text{frontier}}]\}$ (Identify the frontier window)
$\bar{c}_t \leftarrow \frac{1}{|\mathcal{W}_t|} \sum_{i \in \mathcal{W}_t} p^i_\theta(\hat{x}^i_{t+1} \mid \mathbf{x}_t)$ (Compute average confidence)
Let $R_{b;t}$ be the number of masked tokens remaining in block b.
if $\bar{c}_t < \gamma$ **and** $R_{b;t} > N_e$ **and** budgets not exhausted **then**
    return True (Frontier is information-poor)
else
    return False

Algorithm B.3.6 TARGETEDEXPLORE Subroutine

Input: $\mathbf{x}_{t+1}, b, \alpha, \beta, C_{\text{info}}$
Construct $\mathcal{H}_t$ using Eq. (3.6).
for $j \in \mathcal{H}_t$ **do**
    $\mathbf{x}_{t+1;j} \leftarrow \mathbf{x}_{t+1}$, $x^j_{t+1;j} = \arg\max_{v \in \mathcal{V}} p^j_\theta(v \mid \mathbf{x}_t)$. (Form the search beam)
Compute $(\mathbf{x}_{t+2;j})_{j \in \mathcal{H}_t} \leftarrow \text{EXPLOIT}((\mathbf{x}_{t+1;j})_{j \in \mathcal{H}_t}, b, C)$ in batch.
Select $j^\star \leftarrow \arg\max_j s(j)$ for the score in Eq. (3.7).
return $\mathbf{x}_{t+2;j^*}$

Appendix C

Appendix for Theoretical Foundations for Statistical Watermarking

C.1 Proofs

C.1.1 Proof of Theorem 4.4.2

Proof. Let ρ' denote the marginal probability of X and let η denote the marginal probability of R. In the bound of Type I error, choosing $\pi = \delta_y$ yields

$$\begin{aligned}\alpha &\geq \mathbb{P}_{X\sim\pi, R\sim\mathcal{P}(\Omega,\cdot)}(X \in R) \\ &= \mathbb{P}_{R\sim\eta}(y \in R) \\ &= \sum_{R\in 2^\Omega}\left(\sum_{x\in\Omega}\rho'(x)\mathcal{P}(R|x)\right)\cdot \mathbb{1}(y\in R). \end{aligned} \tag{C.1}$$

Now notice that

$$\begin{aligned}\mathcal{P}(X\in R) &= \mathbb{E}_{\mathcal{P}}[\mathbb{1}(X\in R)] \\ &= \sum_{y\in\Omega}\sum_{R\in 2^\Omega}\rho'(y)\mathcal{P}(R|y)\mathbb{1}(y\in R) \\ &= \sum_{y\in\Omega}\underbrace{\left(\sum_{R\in 2^\Omega}\rho'(y)\mathcal{P}(R|y)\cdot\mathbb{1}(y\in R)\right)}_{A(y)}.\end{aligned}$$

For the term $A(y)$, we first know that $A(y)\leq\rho'(y)$. Applying Eq. (C.1), we further have

$$\begin{aligned}A(y) &\leq \sum_{R\in 2^\Omega}\left(\sum_{x\in\Omega}\rho'(x)\mathcal{P}(R|x)\right)\cdot\mathbb{1}(y\in R) \\ &\leq \alpha.\end{aligned}$$

Combining the above two inequalies, it follows that

$$\begin{aligned}\mathcal{P}(X \in R) &\leq \sum_{y\in\Omega} (\alpha \wedge \rho'(y)) \\ &= 1 - \sum_{x\in\Omega:\rho'(x)>\alpha} (\rho'(x) - \alpha) \\ &\leq 1 - \min_{\mathtt{TV}(\rho'\|\rho)\leq\epsilon} \sum_{x\in\Omega:\rho'(x)>\alpha} (\rho'(x) - \alpha) \\ &\leq 1 - \left(\sum_{x\in\Omega:\rho(x)>\alpha} (\rho(x) - \alpha) - \epsilon \right)_+\end{aligned}$$

where the first equality is achieved by

$$\rho' = \arg \min_{\mathtt{TV}(\rho'\|\rho)\leq\epsilon} \sum_{x\in\Omega:\rho'(x)>\alpha} (\rho'(x) - \alpha)$$

and the third inequality is achieved when $\sum_{x\in\Omega:\rho(x)<\alpha} (\alpha - \rho(x)) \geq \epsilon$, a sufficient condition for which being $|\Omega| \geq (2+\epsilon)/\alpha$ (indeed, otherwise we have $1 \geq \sum_{x:\rho(x)<\alpha} \rho(x) > \alpha \cdot |\{x : \rho(x) < \alpha\}| - \epsilon \geq \alpha \cdot (|\Omega| - 1/\alpha) - \epsilon = 1$, a contradiction). This establishes the optimal Type II error.

Finally, to verify that $\mathcal{P}^*$ satisfies the conditions, the condition $\mathtt{TV}(\mathcal{P}^*(\cdot, 2^\Omega)\|\rho) \leq \epsilon$ is clearly satisfied. For any $y \in \Omega$ we have

$$\begin{aligned}\mathbb{P}_{R\sim\eta}(y \in R) &= \sum_{x\in\Omega} \rho^*(x) \cdot \mathbb{P}(R = \{x\}) \cdot \mathbb{1}(y = x) \\ &= \rho^*(y) \cdot \left(1 \wedge \frac{\alpha}{\rho^*(y)}\right) \\ &\leq \alpha.\end{aligned}$$

This implies the $\sup_{\pi\in\Delta(\Omega)} \mathbb{P}_{Y\sim\pi,(X,R)\sim\mathcal{P}^*}(Y \in R) \leq \alpha$ because any π can be written as linear combination of δ_y. Moreover,

$$\begin{aligned}\mathcal{P}^*(X \in R) &= \sum_{x\in\Omega} \rho^*(x) \cdot \mathbb{P}(R = \{x\}) \\ &= \sum_{y\in\Omega} (\alpha \wedge \rho^*(y)) \\ &= 1 - \sum_{x\in\Omega:\rho^*(x)>\alpha} (\rho^*(x) - \alpha)\,.\end{aligned}$$

This verifies that ρ^* achieves the advertised Type II error. □

C.1.2 Proof of Theorem 4.4.7

Proof. **Lower bound.** By definition of Type I error, for any level-α model-agnostic watermarking $(\eta, \{\mathcal{P}_\rho\}_{\rho\in\Delta(\Omega,\mathcal{F})})$, the following holds

$$\sum_{A\in 2^\Omega} \eta(A)\mathbb{1}(x \in A) \leq \alpha,\ \forall x \in \Omega.$$

For simplicity of notations, we assume $\Omega = \{1, 2, \ldots, m\}$. We consider hard instances in the form of $\mathrm{Unif}(i_1, i_2, \ldots, i_{1/\alpha})$ where $1 \le i_1 < \cdots < i_{1/\alpha} \le m$. Notice that for any $\rho_0 = \mathrm{Unif}(i_1, i_2, \ldots, i_{1/\alpha})$, we have $\beta(\mathcal{P}^*_{\rho_0}) = 0$ and

$$\begin{aligned}\beta(\mathcal{P}_{\rho_0}) &\ge \mathbb{P}_{A\sim\eta}\left(\{i_1, \ldots, i_{1/\alpha}\} \cap A = \emptyset\right) \\ &\ge \sum_A \eta(A) \cdot \prod_{j=1}^{1/\alpha} \mathbb{1}(i_j \notin A).\end{aligned}$$

By the probabilistic method,

$$\begin{aligned}\max_{\rho_0} \beta(\mathcal{P}_{\rho_0}) &\ge \max_{i_1<\cdots<i_{1/\alpha}} \sum_A \eta(A) \cdot \prod_{j=1}^{1/\alpha} \mathbb{1}(i_j \notin A) \\ &\ge \frac{1}{\binom{m}{1/\alpha}} \sum_{i_1<\cdots<i_{1/\alpha}} \sum_A \eta(A) \cdot \prod_{j=1}^{1/\alpha} \mathbb{1}(i_j \notin A).\end{aligned}$$

It follows that the maximum excess Type II error is lower bounded by the optimum v^* of the following linear program

$$\begin{aligned}v^* = \min_{\eta}\ & \frac{1}{\binom{m}{1/\alpha}} \sum_{i_1<\cdots<i_{1/\alpha}} \sum_A \eta(A) \cdot \prod_{j=1}^{1/\alpha} \mathbb{1}(i_j \notin A) \\ \text{s.t.}\ & \sum_{A\in 2^\Omega} \eta(A)\mathbb{1}(x \in A) \le \alpha,\ \forall x \in \Omega, \\ & \sum_{A\in 2^\Omega} \eta(A) = 1,\ \eta(A) \ge 0,\ \forall A \in 2^\Omega.\end{aligned}$$

By duality, we have $v^* =$

$$\begin{aligned}&\min_{\eta\ge 0} \max_{\xi,\zeta\ge 0} \frac{1}{\binom{m}{1/\alpha}} \sum_{i_1<\cdots<i_{1/\alpha}} \sum_A \eta(A) \cdot \prod_{j=1}^{1/\alpha} \mathbb{1}(i_j \notin A) + \sum_x \xi(x) \left(\sum_{A\in 2^\Omega} \eta(A)\mathbb{1}(x \in A) - \alpha\right) \\ &\quad + \zeta \cdot \left(\sum_{A\in 2^\Omega} \eta(A) - 1\right)\Bigg) \\ &\ge \max_{\xi,\zeta\ge 0} \min_{\eta\ge 0} \frac{1}{\binom{m}{1/\alpha}} \left(\sum_A \eta(A) \cdot \left(\sum_{i_1<\cdots<i_{1/\alpha}} \prod_{j=1}^{1/\alpha} \mathbb{1}(i_j \notin A) + \sum_x \xi(x)\mathbb{1}(x \in A) + \zeta\right) - \alpha \cdot \sum_x \xi(x) - \zeta\right) \\ &\ge \min_{\eta\ge 0} \frac{1}{\binom{m}{1/\alpha}} \sum_{l=1}^{m} \sum_{|A|=l} \eta(A) \cdot \binom{m-l}{1/\alpha} + l \cdot \xi^* + \zeta^*\Bigg) - \frac{\alpha m \xi^* + \zeta^*}{\binom{m}{1/\alpha}},\end{aligned}$$

where $\xi^* = \binom{m-\alpha m-1}{1/\alpha-1}$ and $\zeta^* = 0$.

Define $f(l) := \binom{m-l}{1/\alpha} + l \cdot \xi^* + \zeta^*$. Since the binomial coefficient $\binom{m-l}{1/\alpha}$ is convex and $l \cdot \xi^* + \zeta^*$ is linear, f a convex function. Notice that

$$\begin{aligned} f(\alpha m) - f(\alpha m - 1) &= \xi^* - \binom{m - \alpha m}{1/\alpha - 1} \\ &< 0 \end{aligned}$$

and

$$\begin{aligned} f(\alpha m + 1) - f(\alpha m) &= \xi^* - \binom{m - \alpha m - 1}{1/\alpha - 1} \\ &= 0, \end{aligned}$$

f achieves minimum at $l^* = \alpha m$. It follows that

$$\binom{m-l}{1/\alpha} + l \cdot \xi^* + \zeta^* \geq \binom{m - \alpha m}{1/\alpha} + \alpha m \cdot \binom{m - \alpha m - 1}{1/\alpha - 1}$$

holds for all $l \in [m]$ and thus

$$\textbf{RHS} \geq \frac{\binom{m-\alpha m}{1/\alpha}}{\binom{m}{1/\alpha}} = \frac{\binom{m-1/\alpha}{\alpha m}}{\binom{m}{\alpha m}} = \frac{\binom{m-\frac{1}{\alpha}}{\alpha m}}{\binom{m}{\alpha m}}.$$

Upper bound. Define p as the projection from $\Omega \times 2^\Omega$ to 2^Ω, i.e. $p(V) = \{A \in 2^\Omega : \exists x \in \Omega,\ s.t. (x, A) \in V\}$. Let $W := \{(x, A) \in \Omega \times 2^\Omega : x \in A\}$.

Notice that the marginal distribution of reject region

$$\eta^*(A) = \begin{cases} \frac{1}{\binom{m}{\alpha m}} & \text{if } |A| = \alpha m \\ 0 & \text{otherwise.} \end{cases}$$

satisfies

$$\begin{aligned} \sup_{\pi \in \Delta(\Omega)} \mathbb{P}_{Y \sim \pi, (X,R) \sim \mathcal{P}}(Y \in R) &\leq \sum_{x \in \Omega} \pi(x) \cdot \eta^* \left(p(W \cap (\{x\} \times 2^\Omega)) \right) \\ &\leq \sum_{x \in \Omega} \pi(x) \cdot \left(1 - \frac{\binom{m-1}{\alpha m}}{\binom{m}{\alpha m}} \right) \\ &= \alpha. \end{aligned}$$

Hence it guarantees Type I error $\leq \alpha$.

It suffices to show **(*)**: for any $\rho \in \Delta(\Omega, \mathcal{F})$, there exists a coupling $\mathcal{P}_\rho$ of η^* and ρ such that $\mathbb{P}_{(x,A) \sim \mathcal{P}_\rho}(x \notin A) \leq \frac{\binom{m-\frac{1}{\alpha}}{\alpha m}}{\binom{m}{\alpha m}} + \sum_{x: \rho(x) \geq \alpha} (\rho(x) - \alpha)$.

To show the above, we need to check Strassen's condition

$$\rho(U) - \eta^* \left(p \left(W \cap (U \times 2^\Omega) \right) \right) \leq \frac{\binom{m-\frac{1}{\alpha}}{\alpha m}}{\binom{m}{\alpha m}} + \sum_{x: \rho(x) \geq \alpha} (\rho(x) - \alpha),\ \forall U \subset \Omega. \tag{C.2}$$

Indeed, given Eq. (C.2), Theorem 11 in Strassen [183] establishes **(*)**.

In the rest of the proof, we show Eq. (C.2). Fix U with cardinality k. First notice that

$$\rho(U) - \sum_{x:\rho(x)\geq\alpha} (\rho(x) - \alpha) \leq (\alpha k \wedge 1).$$

Since $p\left(W \cap (U \times 2^\Omega)\right) = \{A \in 2^\Omega : \exists i \in U,\ s.t.\ i \in A\}$, we have

$$\eta^*\left(p\left(W \cap (U \times 2^\Omega)\right)\right) \geq 1 - \frac{\binom{m-k}{\alpha m}}{\binom{m}{\alpha m}} = 1 - \frac{\binom{m-\alpha m}{k}}{\binom{m}{k}}.$$

In the remaining paper, $\binom{m-\alpha m}{k}$ is understood as zero if $m - \alpha m < k$.

If $k \leq \frac{1}{\alpha}$, then because $g(k) := \alpha k - 1 + \frac{\binom{m-\alpha m}{k}}{\binom{m}{k}}$ is convex and takes maximum $\frac{\binom{m-\alpha m}{\frac{1}{\alpha}}}{\binom{m}{\frac{1}{\alpha}}} = \frac{\binom{m-\frac{1}{\alpha}}{\alpha m}}{\binom{m}{\alpha m}}$ at $k^* = \frac{1}{\alpha}$, we have

$$\begin{aligned}\rho(U) - \eta^*\left(p\left(W \cap (U \times 2^\Omega)\right)\right) &\leq \alpha k - 1 + \frac{\binom{m-\alpha m}{k}}{\binom{m}{k}} + \sum_{x:\rho(x)\geq\alpha} (\rho(x) - \alpha) \\ &= \frac{\binom{m-\frac{1}{\alpha}}{\alpha m}}{\binom{m}{\alpha m}} + \sum_{x:\rho(x)\geq\alpha} (\rho(x) - \alpha).\end{aligned}$$

If $k \geq \frac{1}{\alpha}$, then since $\frac{\binom{m-\alpha m}{k}}{\binom{m}{k}} = \frac{\binom{m-k}{\alpha m}}{\binom{m}{\alpha m}}$ is monotonously decreasing in k,

$$\begin{aligned}\rho(U) - \eta^*\left(p\left(W \cap (U \times 2^\Omega)\right)\right) &\leq \frac{\binom{m-\alpha m}{k}}{\binom{m}{k}} + \sum_{x:\rho(x)\geq\alpha} (\rho(x) - \alpha) \\ &= \frac{\binom{m-\frac{1}{\alpha}}{\alpha m}}{\binom{m}{\alpha m}} + \sum_{x:\rho(x)\geq\alpha} (\rho(x) - \alpha).\end{aligned}$$

Combining, we establish Eq. (C.2).

Under the condition $\alpha \to 0_+$ and $1/(\alpha m) \to 0_+$, the rate displayed in Theorem 1 simplifies to:

$$\frac{(m - \alpha m)(m - \alpha m - 1) \cdots (m - \alpha m - 1/\alpha + 1)}{n(m-1)\cdots(m - 1/\alpha + 1)} \asymp (1-\alpha)^{1/\alpha} \to e^{-1}.$$

This concludes the proof. □

C.1.3 Proof of Theorem 4.4.10

Proof. We follow the notation in the proof of Theorem 4.4.7.

Lower bound. By definition of Type I error, for any level-α model-agnostic watermarking $(\eta, \{\mathcal{P}_\rho\}_{\rho\in\Delta(\Omega,\mathcal{F})})$, the following holds

$$\sum_{A\in 2^\Omega} \eta(A)\mathbb{1}(x \in A) \leq \alpha,\ \forall x \in \Omega.$$

We consider hard instances in the form of $\mathrm{Unif}(i_1, i_2, \ldots, i_{1/\kappa})$ where $1 \le i_1 < \cdots < i_{1/\kappa} \le m$. Notice that for any $\rho_0 = \mathrm{Unif}(i_1, i_2, \ldots, i_{1/\kappa})$, we have $\beta(\mathcal{P}^*_{\rho_0}) = 0$ and

$$\begin{aligned}\beta(\mathcal{P}_{\rho_0}) &\ge \mathbb{P}_{A\sim\eta}\left(\{i_1, \ldots, i_{1/\kappa}\} \cap A = \emptyset\right)\\ &\ge \sum_A \eta(A) \cdot \prod_{j=1}^{1/\kappa} \mathbb{1}(i_j \notin A).\end{aligned}$$

By the probabilistic method,

$$\begin{aligned}\max_{\rho_0} \beta(\mathcal{P}_{\rho_0}) &\ge \max_{i_1<\cdots<i_{1/\kappa}} \sum_A \eta(A) \cdot \prod_{j=1}^{1/\kappa} \mathbb{1}(i_j \notin A)\\ &\ge \frac{1}{\binom{m}{1/\kappa}} \sum_{i_1<\cdots<i_{1/\kappa}} \sum_A \eta(A) \cdot \prod_{j=1}^{1/\kappa} \mathbb{1}(i_j \notin A).\end{aligned}$$

It follows that the maximum excess Type II error is lower bounded by the optimum v^* of the following linear program

$$\begin{aligned}v^* = \min_{\eta}\ & \frac{1}{\binom{m}{1/\kappa}} \sum_{i_1<\cdots<i_{1/\kappa}} \sum_A \eta(A) \cdot \prod_{j=1}^{1/\kappa} \mathbb{1}(i_j \notin A)\\ \text{s.t.}\ & \sum_{A\in 2^\Omega} \eta(A)\mathbb{1}(x \in A) \le \alpha,\ \forall x \in \Omega,\\ & \sum_{A\in 2^\Omega} \eta(A) = 1,\ \eta(A) \ge 0,\ \forall A \in 2^\Omega.\end{aligned}$$

By duality, we have $v^* =$

$$\begin{aligned}&\min_{\eta\ge 0} \max_{\xi,\zeta\ge 0} \frac{1}{\binom{m}{1/\kappa}} \sum_{i_1<\cdots<i_{1/\kappa}} \sum_A \eta(A) \cdot \prod_{j=1}^{1/\kappa} \mathbb{1}(i_j \notin A) + \sum_x \xi(x) \left(\sum_{A\in 2^\Omega} \eta(A)\mathbb{1}(x \in A) - \alpha\right)\\ &+ \zeta \cdot \left(\sum_{A\in 2^\Omega} \eta(A) - 1\right)\Bigg)\\ =\ &\max_{\xi,\zeta\ge 0} \min_{\eta\ge 0} \frac{1}{\binom{m}{1/\kappa}} \left(\sum_A \eta(A) \cdot \left(\sum_{i_1<\cdots<i_{1/\kappa}} \prod_{j=1}^{1/\kappa} \mathbb{1}(i_j \notin A) + \sum_x \xi(x)\mathbb{1}(x \in A) + \zeta\right) - \alpha \cdot \sum_x \xi(x) - \zeta\right)\\ \ge\ &\min_{\eta\ge 0} \frac{1}{\binom{m}{1/\kappa}} \sum_{l=1}^{m} \sum_{|A|=l} \eta(A) \cdot \binom{m-l}{1/\kappa} + l \cdot \xi^* + \zeta^*\Bigg) - \frac{\alpha m \xi^* + \zeta^*}{\binom{m}{1/\kappa}},\end{aligned}$$

where $\xi^* = \binom{m-\alpha m-1}{1/\kappa-1}$ and $\zeta^* = 0$.

Define $f(l) := \binom{m-l}{1/\kappa} + l \cdot \xi^* + \zeta^*$. Since the binomial coefficient $\binom{m-l}{1/\kappa}$ is convex and $l \cdot \xi^* + \zeta^*$ is linear, f a convex function. Notice that

$$\begin{aligned}f(\alpha m) - f(\alpha m - 1) &= \xi^* - \binom{m-\alpha m}{1/\kappa-1}\\ &< 0\end{aligned}$$

and

$$\begin{aligned} f(\alpha m+1)-f(\alpha m) &= \xi^* - \binom{m-\alpha m-1}{1/\kappa-1} \\ &= 0, \end{aligned}$$

so f achieves its minimum at $l^* = \alpha m$. It follows that

$$\binom{m-l}{1/\kappa} + l\cdot\xi^* + \zeta^* \geq \binom{m-\alpha m}{1/\kappa} + \alpha m \cdot \binom{m-\alpha m-1}{1/\kappa-1}$$

holds for all $l \in [m]$ and thus

$$\textbf{RHS} \geq \frac{\binom{m-\alpha m}{1/\kappa}}{\binom{m}{1/\kappa}} = \frac{\binom{m-1/\kappa}{\alpha m}}{\binom{m}{\alpha m}} = \frac{\binom{m-\frac{1}{\kappa}}{\alpha m}}{\binom{m}{\alpha m}}.$$

Upper bound. Notice that the marginal distribution of reject region

$$\eta^*(A) = \begin{cases} \frac{1}{\binom{m}{\alpha m}} & \text{if } |A| = \alpha m \\ 0 & \text{otherwise.} \end{cases}$$

satisfies

$$\begin{aligned} \sup_{\pi\in\Delta(\Omega)} \mathbb{P}_{Y\sim\pi,(X,R)\sim\mathcal{P}}(Y\in R) &\leq \sum_{x\in\Omega}\pi(x)\cdot\eta^*\left(p(W\cap(\{x\}\times 2^\Omega))\right) \\ &\leq \sum_{x\in\Omega}\pi(x)\cdot \quad 1-\frac{\binom{m-1}{\alpha m}}{\binom{m}{\alpha m}}\Bigg) \\ &= \alpha. \end{aligned}$$

Hence we have a guarantee on Type I error $\leq \alpha$.

In what remains, we define p and W in the same way in the proof of Theorem 4.4.7 and check Strassen's condition

$$\rho(U) - \eta^*\left(p\left(W\cap(U\times 2^\Omega)\right)\right) \leq \frac{\binom{m-\frac{1}{\alpha}}{\alpha m}}{\binom{m}{\alpha m}} + \sum_{x:\rho(x)\geq\alpha}(\rho(x)-\alpha),\ \forall U\subset\Omega. \tag{C.3}$$

Fix U with cardinality k. Due to the condition of $\sup_{\omega\in\Omega}\rho(\{\omega\}) \leq \kappa$, we have $\rho(U) - \sum_{x:\rho(x)\geq\alpha}(\rho(x)-\alpha) \leq (\kappa k \wedge 1)$. Since $p\left(W\cap(U\times 2^\Omega)\right) = \{A\in 2^\Omega : \exists i\in U,\ s.t.\ i\in A\}$, we have

$$\eta^*\left(p\left(W\cap(U\times 2^\Omega)\right)\right) \geq 1 - \frac{\binom{m-k}{\alpha m}}{\binom{m}{\alpha m}} = 1 - \frac{\binom{m-\alpha m}{k}}{\binom{m}{k}}.$$

If $k \leq \frac{1}{\kappa}$, then

$$\begin{aligned} \rho(U) - \eta^*\left(p\left(W\cap(U\times 2^\Omega)\right)\right) &\leq \kappa k - 1 + \frac{\binom{m-\alpha m}{k}}{\binom{m}{k}} + \sum_{x:\rho(x)\geq\alpha}(\rho(x)-\alpha) \\ &= \frac{\binom{m-\frac{1}{\kappa}}{\alpha m}}{\binom{m}{\alpha m}} + \sum_{x:\rho(x)\geq\alpha}(\rho(x)-\alpha), \end{aligned}$$

where the second step follows from the fact that $g(k) := \kappa k - 1 + \frac{\binom{m-\alpha m}{k}}{\binom{m}{k}}$ is convex and takes maximum $\frac{\binom{m-\alpha m}{\frac{1}{\kappa}}}{\binom{m}{\frac{1}{\kappa}}} = \frac{\binom{m-\frac{1}{\kappa}}{\alpha m}}{\binom{m}{\alpha m}}$ at $k^* = \frac{1}{\kappa}$.

If $k \geq \frac{1}{\kappa}$, then

$$\begin{aligned}\rho(U) - \eta^*\left(p\left(W \cap (U \times 2^\Omega)\right)\right) &\leq \frac{\binom{m-\alpha m}{k}}{\binom{m}{k}} + \sum_{x:\rho(x)\geq\alpha}(\rho(x)-\alpha) \\ &= \frac{\binom{m-\frac{1}{\kappa}}{\alpha m}}{\binom{m}{\alpha m}} + \sum_{x:\rho(x)\geq\alpha}(\rho(x)-\alpha),\end{aligned}$$

where the inequality is because $\frac{\binom{m-\alpha m}{k}}{\binom{m}{k}} = \frac{\binom{m-k}{\alpha m}}{\binom{m}{\alpha m}}$ is monotonously decreasing in k. Combining the two cases establishes Eq. (C.3).

Combining the above cases, we checked Strassen's condition and hence the statement follows. □

C.1.4 Proof of Theorem 4.4.11

Proof. Throughout the proof we assume that $h < 1/4$, otherwise the bounds become trivial.

We first prove the lower bound. For this purpose, we construct the hard instance: let $q_0 = H_b^{-1}(h)$ (take the one $\geq 1/2$) where H_b is the binary entropy function defined by $H_b(x) = -x\ln x - (1-x)\ln(1-x)$, and set $\rho_0 = (1-q_0)\delta_{x_1} + q_0\delta_{x_2}$ where x_1, x_2 are two different elements in Ω_0. Then Lemma C.1.2 implies that $q_0 \geq 3/4$. By Theorem 4.4.2,

$$\begin{aligned}\beta = 1 - \mathcal{P}(X \in R) &= \sum_{x\in\Omega:\rho(x)>\alpha}(\rho(x)-\alpha) \\ &\geq \frac{1}{2}\cdot\mathbb{P}\left(\rho(X) \geq 2\alpha\right) \\ &= \frac{1}{2}\cdot\mathbb{P}\left(\sum_{i=1}^n \ln\rho_0(X_i) \geq \ln(2\alpha)\right) \\ &\geq \mathbb{1}(n\ln q_0 \geq \ln(2\alpha))\cdot\frac{1}{2}q_0^n \\ &\geq \mathbb{1}(2n(1-q_0) \leq -\ln(2\alpha))\cdot\frac{1}{2}\exp\left(-2n(1-q_0)\right) \\ &\geq \mathbb{1}\left(n \leq \frac{\ln\frac{1}{2\alpha}}{2h/\ln\frac{\ln 2}{h}}\right)\cdot\frac{1}{2}\exp\left(-\frac{2nh}{\ln\frac{\ln 2}{h}}\right),\end{aligned}$$

where the last inequality follows from Lemma C.1.3. It follows that

$$n(h,\alpha,\beta) \geq \frac{\ln\frac{\ln 2}{h}}{2h}\cdot\left(\ln\frac{1}{2\alpha}\wedge\ln\frac{1}{2\beta}\right). \tag{C.4}$$

Furthermore, suppose $n \leq \frac{\ln \frac{1}{2\alpha}}{4(1-q_0)\ln \frac{1}{1-q_0}}$. Define $Y = \sum_{i=1}^{n} \mathbb{1}(\rho_0(X_i) = 1 - q_0)$, then notice that $Y \sim \text{Binom}(n, 1-q_0)$ and if $Y \leq \frac{\ln \frac{1}{2\alpha}}{2\ln \frac{1}{1-q_0}}$, then

$$\begin{aligned}\sum_{i=1}^{n} \ln \rho_0(X_i) &\geq \frac{\ln \frac{1}{2\alpha}}{2\ln \frac{1}{1-q_0}} \cdot \ln(1-q_0) + n \cdot \ln q_0 \\ &\geq \ln(2\alpha),\end{aligned}$$

where the last inequality is due to $n \cdot \ln q_0 \geq -2(1-q_0)n \geq -2(1-q_0)\frac{\ln \frac{1}{2\alpha}}{4(1-q_0)\ln \frac{1}{1-q_0}} = \frac{\ln(2\alpha)}{2\ln \frac{1}{1-q_0}} \geq \frac{\ln(2\alpha)}{2}$. Applying this and Markov's inequality,

$$\begin{aligned}\mathbb{P}\left(\sum_{i=1}^{n} \ln \rho_0(X_i) \geq \ln(2\alpha)\right) &\geq \mathbb{P}\left(Y \leq \frac{2\ln \frac{1}{2\alpha}}{\ln \frac{1}{1-q_0}}\right) \\ &\geq 1 - \frac{n(1-q_0)}{\frac{2\ln \frac{1}{2\alpha}}{\ln \frac{1}{1-q_0}}} \\ &\geq \frac{1}{2},\end{aligned}$$

which is a contradiction to $\mathbb{P}(\rho(X) \geq 2\alpha) \leq 2\beta$. As a result,

$$\begin{aligned}n(h, \alpha, \beta) &\geq \frac{\ln \frac{1}{2\alpha}}{4(1-q_0)\ln \frac{1}{1-q_0}} \\ &\geq \frac{\ln \frac{1}{2\alpha}}{4h}.\end{aligned} \tag{C.5}$$

Combining Eq. (C.4) and Eq. (C.5), we establish the lower bound.

For the upper bound, we define $q = \max_{x \in \Omega_0} \rho_0(x)$, then Lemma C.1.2 implies that $q \geq 1/2$. Define $Y = \sum_{i=1}^{n} \mathbb{1}(\rho_0(X_i) \neq q)$ (recall that $Y \sim \text{Binom}(n, 1-q)$). It suffices to show when

$$n = 900 \left(\frac{2\ln \frac{9k}{h}}{h} \cdot \left(\ln \frac{1}{\alpha} \wedge \ln \frac{1}{\beta}\right)\right) \vee \frac{(18 + 36\ln(9k))\ln \frac{1}{\alpha}}{h}$$

the Type II error of the UMP watermark $1 - \mathcal{P}^*(X \in R) \leq \beta$.

By Theorem 4.4.2 and Bennett's inequality,

$$\begin{aligned}
1-\mathcal{P}^*(X\in\mathbb{R}) &= \sum_{x\in\Omega:\rho(x)>\alpha}(\rho(x)-\alpha)\\
&\le \mathbb{P}\left(\rho(X)\ge\alpha\right)\\
&= \mathbb{P}\left(\sum_{i=1}^{n}\ln\rho_0(X_i)\ge\ln(\alpha)\right)\\
&\le \mathbb{P}\left(Y\le\frac{\ln\frac{1}{\alpha}}{\ln\frac{1}{1-q}}\right)\\
&\le \exp\left(-nq(1-q)\theta\left(\frac{1-q-\frac{\ln\frac{1}{\alpha}}{n\ln\frac{1}{1-q}}}{q(1-q)}\right)\right) \qquad \text{(C.6)}
\end{aligned}$$

where $\theta(x)=(1+x)\ln(1+x)-x$; the penultimate inequality follows from $\sum_{i=1}^{n}\ln\rho_0(X_i)\le Y\ln(1-q)$.

Notice that by Lemma C.1.2,

$$\begin{aligned}
(1-q)\ln\frac{1}{1-q} &\ge \frac{h}{9\ln\frac{9k\ln(9k)}{h}}\cdot\ln\frac{\ln\frac{1}{h}}{h}\\
&= h\cdot\frac{\ln\ln\frac{1}{h}+\ln\frac{1}{h}}{9\left(\ln\frac{1}{h}+\ln(9k\ln(9k))\right)}\\
&\ge \frac{h}{9+\ln(9k\ln(9k))}.
\end{aligned}$$

Since $n\ge\frac{(18+36\ln(9k\ln(9k)))\ln\frac{1}{\alpha}}{h}$, we have $n\ge\frac{2}{1-q}\frac{\ln\frac{1}{\alpha}}{\ln\frac{1}{1-q}}$. Under this condition, we have the simplification

$$\begin{aligned}
\theta\left(\frac{1-q-\frac{\ln\frac{1}{\alpha}}{n\ln\frac{1}{1-q}}}{q(1-q)}\right) &\ge \theta\left(\frac{1}{2q}\right)\\
&\ge \frac{1}{50}.
\end{aligned}$$

Plugging back to Eq. (C.6), we have

$$\begin{aligned}
1-\mathcal{P}^*(X\in\mathbb{R}) &\le \exp\left(-nq(1-q)\theta\left(\frac{1-q-\frac{\ln\frac{1}{\alpha}}{n\ln\frac{1}{1-q}}}{q(1-q)}\right)\right)\\
&\le \exp\left(-\frac{n(1-q)}{100}\right)\\
&\le \exp\left(-\frac{nh}{900\ln\frac{9k\ln(9k)}{h}}\right) \qquad \text{(C.7)}
\end{aligned}$$

where we applied Lemma C.1.3 in the last step.

Furthermore, we have

$$\begin{aligned} 1-\mathcal{P}(X\in R) &\le \mathbb{P}\quad \sum_{i=1}^{n}\ln\rho_0(X_i)\ge\ln(\alpha)\Bigg) \\ &\le \mathbb{1}\left(n\le\frac{\ln\alpha}{\ln q}\right) \\ &\le \mathbb{1}\quad n\le 900\quad \frac{\ln\frac{9k\ln(9k)}{h}}{h}\cdot\ln\frac{1}{\alpha}\Bigg)\Bigg), \end{aligned} \tag{C.8}$$

where the last step is due to Lemma C.1.3. Combining Eq. (C.7) and Eq. (C.8), we know that $1-\mathcal{P}^*(X\in\mathbb{R})\le\beta$ when $n\ge 900\left(\frac{\ln\frac{9k\ln(9k)}{h}}{h}\cdot\left(\ln\frac{1}{\alpha}\wedge\ln\frac{1}{\beta}\right)\right)$. This establishes the upper bound. □

Supporting lemmata

Lemma C.1.1 (Topsøe [188], Theorem 1.2)**.** *Define the binary entropy function $H_b:(0,1)\to\mathbb{R}$ as $H_b(x)=-x\ln x-(1-x)\ln(1-x)$. Then $4x(1-x)\le H_b(x)\le(4x(1-x))^{1/\ln 4}$.*

Lemma C.1.2. *Suppose ρ is a probability measure over Ω such that $H(\rho)=h$, and define $q=\max_{x\in\Omega}\rho(x)$. If $H(\rho)\le 1/4$, then $q\ge 1/2$. Furthermore, if $H_b(q)\le 1/4$, then $q\ge 3/4$.*

Proof. Suppose $q\le 1/2$. By convexity of H,

$$H(\rho)\ge-\left\lfloor\frac{1}{q}\right\rfloor q\ln q\ge-\frac{1}{2}\ln\frac{1}{2}\ge 1/4.$$

This is a contradiction.

Suppose $q\le 3/4$, then Lemma C.1.1 implies that

$$H_b(q)\ge 4q(1-q)\ge 1/4.$$

This is a contradiction. □

Lemma C.1.3. *Suppose ρ is a probability measure over Ω such that $H(\rho)=h$ and $|\Omega|=k$. Define $q=\max_{x\in\Omega}\rho(x)$. If $q\ge 1/2$, then we have*

$$\frac{h}{9\ln\frac{9k\ln(9k)}{h}}\le 1-q\le\frac{h}{\ln\frac{\ln 2}{h}}$$

Proof. We have

$$H(\rho)\ge-(1-q)\ln(1-q)\ge(1-q)\cdot\ln 2.$$

It follows that

$$\begin{aligned} h &\ge -(1-q)\ln(1-q) \\ &\ge (1-q)\ln\frac{\ln 2}{h}. \end{aligned}$$

Therefore $1-q \le \frac{h}{\ln \frac{\ln 2}{h}}$.

By the convexity of H and $-q\ln q \le 2(1-q)$,

$$\begin{aligned} H(\rho) \le & -q\ln q - (1-q)\ln\frac{1-q}{k} \\ & \le (1-q)\ln\frac{9k}{1-q}. \end{aligned}$$

This means that

$$\begin{aligned} h^2 &\le (1-q)^2\left(\ln\frac{9k}{1-q}\right)^2 \\ &\le 2(1-q)^2\left(\ln^2(9k)+\ln^2(1-q)\right) \\ &\le 2(1-q)^2\ln^2(9k) + (1-q)\cdot\left(2(1-q)\ln^2(1-q)\right) \\ &\le (1-q)\cdot(\ln^2(9k)+18), \end{aligned} \tag{C.9}$$

where the last inequality is due to $2(1-q)\le 1$ and $2(1-q)\ln^2(1-q)\le 18$. It follows that

$$\begin{aligned} h &\le (1-q)\ln\frac{9k}{1-q} \\ &\le 9(1-q)\ln\frac{9k\ln(9k)}{h}, \end{aligned}$$

where the last step is because $\ln\frac{1}{1-q} \le 2\ln\frac{\ln^2(9k)+18}{h} \le 9\ln\frac{\ln(9k)}{h}$, using Eq. (C.9). This establishes $1-q \ge \frac{h}{9\ln\frac{9k\ln(9k)}{h}}$. □

C.1.5 Proof of Theorem 4.4.16

Proof. Let $\kappa_0 = \alpha/\log\frac{1}{\beta}$, $\kappa := e^{-\widehat{H}} \lesssim \kappa_0$, and $m := |\Omega| = |\Omega_0|^n \gtrsim 1/\kappa_0$. Define p as the projection from $\Omega\times 2^\Omega$ to 2^Ω: $\forall V,\ p(V) = \{A\in 2^\Omega : \exists x\in\Omega,\ s.t.\ (x,A)\in V\}$. Define $\bar\Omega := \{x\in\Omega : \rho(x)\le\kappa\}$, $W := \{(x,A)\in\Omega\times 2^\Omega : x\in A\}$, and $\bar W := \{(x,A)\in\bar\Omega\times 2^\Omega : x\in A\}$. Notice that

$$\begin{aligned} \sup_{\pi\in\Delta(\Omega)} \mathbb{P}_{Y\sim\pi,(X,R)\sim\mathcal{P}}(Y\in R) &\le \sum_{x\in\Omega}\pi(x)\cdot\eta^*\left(p(W\cap(\{x\}\times 2^\Omega))\right) \\ &\le \sum_{x\in\Omega}\pi(x)\cdot\quad 1-\frac{\binom{m-1}{\alpha m}}{\binom{m}{\alpha m}}\Bigg) = \alpha; \end{aligned}$$

thus Type I error $\le\alpha$. To establish the conditional Type II error guarantee, we check the following Strassen's condition, similarly to the proof of Theorem 4.4.7,

$$\rho(U) - \eta^*\left(p\left(\bar W\cap(U\times 2^\Omega)\right)\right) \le \rho(\bar\Omega^c) + \beta\cdot\rho(\bar\Omega),\ \forall U\subset\Omega. \tag{C.10}$$

Fix U and define $k := |U \cap \bar{\Omega}|$. Notice that $\rho(U) \le (\kappa k + \rho(\bar{\Omega}^c)) \wedge 1$. It follows that

$$\begin{aligned}\rho(U) - \eta^* \left(p\left(\bar{W} \cap (U \times 2^\Omega)\right)\right) \le & \left((\kappa k + \rho(\bar{\Omega}^c)) \wedge 1\right) - 1 + \frac{\binom{m-k}{\alpha m}}{\binom{m}{\alpha m}} \\ \le & \max\left\{\frac{\binom{m-\frac{\rho(\bar{\Omega})}{\kappa}}{\alpha m}}{\binom{m}{\alpha m}} - \rho(\bar{\Omega}^c), 0\right\} + \rho(\bar{\Omega}^c),\end{aligned}$$

where the second step follows from the fact that $\left((\kappa k + \rho(\bar{\Omega}^c)) \wedge 1\right) - 1 + \frac{\binom{m-k}{\alpha m}}{\binom{m}{\alpha m}}$ is convex in $[0, \rho(\bar{\Omega})/\kappa]$ and decreasing in $[\rho(\bar{\Omega})/\kappa, \infty]$, and thus the maximum can only be taken at either $k = 0$ or $k = \rho(\bar{\Omega})/\kappa$.

By the conditions of n and κ, some arithmetic shows that

$$\begin{aligned}\max\left\{\frac{\binom{m-\frac{\rho(\bar{\Omega})}{\kappa}}{\alpha m}}{\binom{m}{\alpha m}} - \rho(\bar{\Omega}^c), 0\right\} \le & \max\left\{\frac{\binom{m-\frac{\rho(\bar{\Omega})}{\kappa_0}}{\alpha m}}{\binom{m}{\alpha m}} - \rho(\bar{\Omega}^c), 0\right\} \\ \lesssim & \max\left\{(1-\alpha)^{\rho(\bar{\Omega})/\kappa_0} - 1 + \rho(\bar{\Omega}), 0\right\} \\ \lesssim & \max\left\{\beta^{\rho(\bar{\Omega})} - 1 + \rho(\bar{\Omega}), 0\right\} \\ \le & \beta \cdot \rho(\bar{\Omega}),\end{aligned}$$

where the first inequality comes from $\kappa \le \kappa_0$, the second inequality is because

$$\begin{aligned}\frac{\binom{m-\frac{\rho(\bar{\Omega})}{\kappa_0}}{\alpha m}}{\binom{m}{\alpha m}} &= \frac{(m-\alpha m)(m-\alpha m-1)\cdots(m-\alpha m-\frac{\rho(\bar{\Omega})}{\kappa_0}+1)}{m(m-1)\cdots(m-\frac{\rho(\bar{\Omega})}{\kappa_0}+1)} \\ &\lesssim (1-\alpha)^{\rho(\bar{\Omega})/\kappa_0}\end{aligned}$$

due to $m \gtrsim 1/\kappa_0$, the third inequality follows from $\kappa_0 = \alpha/\log(1/\beta)$, and the last inequality is due to the observation that $\max\left\{\beta^{\rho(\bar{\Omega})} - 1 + \rho(\bar{\Omega}), 0\right\} - \beta \cdot \rho(\bar{\Omega})$ is a convex function of $\rho(\bar{\Omega})$ and takes maximum at either $\rho(\bar{\Omega}) = 0$ or 1.

Eq. (C.10) hence follows. Applying Strassen's Theorem [183], we have $\mathbb{P}_{(X,R)\sim\mathcal{P}}(\bar{W}) \ge 1 - \left(\rho(\bar{\Omega}^c) + \beta \cdot \rho(\bar{\Omega})\right) = \rho(\bar{\Omega}) \cdot (1-\beta)$.

By Bayes' theorem, $\mathbb{P}_{(X,R)\sim\mathcal{P}}(X \notin R| - \log \rho(X) \ge \widehat{H}) = \mathbb{P}_{(X,R)\sim\mathcal{P}}(X \notin R|\rho(X) \le \kappa) = \frac{1-\mathbb{P}_{(X,R)\sim\mathcal{P}}(\bar{W})}{\rho(\bar{\Omega})} \le \beta$. This completes the proof. □

C.1.6 Proof of Theorem 4.4.20

Proof. Throughout the proof we omit the subscript in the shrinkage operator $\mathcal{S}$, as G is fixed. First notice that

$$\begin{aligned}\mathbb{E}_{X,R\sim\mathcal{P}}\left[\min_{Y\in out(X)} \mathbb{1}(Y \in R)\right] &= \mathcal{P}(X \in \mathcal{S}(R)) \\ &= \sum_{y\in\Omega}\sum_{R\in 2^\Omega} \rho(y)\mathcal{P}(R|y)\mathbb{1}(y \in \mathcal{S}(R)).\end{aligned}$$

Further, notice that $y \in in(z)$ and $y \in \mathcal{S}(R)$ implies that $z \in R$, thus

$$\begin{aligned}\sum_{y\in in(z)}\sum_{R\in 2^\Omega}\rho(y)\mathcal{P}(R|y)\mathbb{1}(y\in\mathcal{S}(R)) &\le \sum_{y\in in(z)}\sum_{R\in 2^\Omega}\rho(y)\mathcal{P}(R|y)\mathbb{1}(z\in R)\\ &\le \sum_{y\in \Omega}\sum_{R\in 2^\Omega}\rho(y)\mathcal{P}(R|y)\mathbb{1}(z\in R)\\ &= \mathbb{P}_{X\sim\delta_z, R\sim\mathcal{P}(\Omega,\cdot)}(X\in R)\\ &\le \alpha.\end{aligned}$$

It follows that the optimum Type II error is lower bounded by the optimum of the following linear program

$$\begin{aligned}&\min_{\mathcal{P}}\ 1-\sum_{y\in\Omega}\sum_{R\in 2^\Omega}\rho(y)\mathcal{P}(R|y)\mathbb{1}(y\in\mathcal{S}(R)) \qquad\qquad (C.11)\\ &s.t.\ \sum_{y\in in(z)}\sum_{R\in 2^\Omega}\rho(y)\mathcal{P}(R|y)\mathbb{1}(y\in\mathcal{S}(R))\le\alpha,\ \sum_{R\in 2^\Omega}\mathcal{P}(R|z)=1, 0\le\mathcal{P}(R|z)\le 1,\ \forall z\in\Omega, R\in 2^\Omega.\end{aligned}$$

We claim that the minimum in Eq. (C.11) is equal to the minimum of Eq. (4.2). Indeed, it suffices to show that Eq. (C.11) is optimized when $\mathcal{P}(\cdot|y_0)$ is supported on $\{\emptyset, \mathcal{S}^{-1}(\{y_0\})$ (then setting $x(y)\equiv\mathcal{P}(\mathcal{S}^{-1}(\{y\})|y)$ reduces Eq. (C.11) to Eq. (4.2)). To see this, consider any minimizer $\widetilde{\mathcal{P}}$ such that there exists $y_0\in\Omega$ and $R_0\notin\{\emptyset,\mathcal{S}^{-1}(\{y_0\})$, with $\widetilde{\mathcal{P}}(R_0|y_0)>0$. We will show that there exists $\bar{\mathcal{P}}$ such that it achieves the no greater objective value, and satisfies $|\text{supp}(\bar{\mathcal{P}}(\cdot|y_0))\cap\{\emptyset,\mathcal{S}^{-1}(\{y_0\})^c|=|\text{supp}(\widetilde{\mathcal{P}}(\cdot|y_0))\cap\{\emptyset,\mathcal{S}^{-1}(\{y_0\})^c|-1$ and $|\text{supp}(\bar{\mathcal{P}}(\cdot|y))|=|\text{supp}(\widetilde{\mathcal{P}}(\cdot|y))|$ for all other $y\in\Omega$. Iteratively applying this argument, we reduce $\text{supp}(\widetilde{\mathcal{P}}(\cdot|y))\cap\{\emptyset,\mathcal{S}^{-1}(\{y\})^c$ to $\emptyset$ for any $y\in\Omega$ and thereby prove the claim.

Consider the following two cases.

Case 1: $y_0\notin\mathcal{S}(R_0)$**.** Then letting

$$\bar{\mathcal{P}}(R|y)=\begin{cases}\widetilde{\mathcal{P}}(R_0|y)+\widetilde{\mathcal{P}}(R|y), & y=y_0, R=\emptyset\\ 0, & y=y_0, R=R_0\\ \widetilde{\mathcal{P}}(R|y), & \text{o.w.},\end{cases}$$

we observe that

$$\sum_{y\in\Omega}\sum_{R\in 2^\Omega}\rho(y)\widetilde{\mathcal{P}}(R|y)\mathbb{1}(y\in\mathcal{S}(R))=\sum_{y\in\Omega}\sum_{R\in 2^\Omega}\rho(y)\bar{\mathcal{P}}(R|y)\mathbb{1}(y\in\mathcal{S}(R))$$

and $\bar{\mathcal{P}}$ satisfies all the constraints in Eq. (C.11). It is obvious from the construction of $\bar{\mathcal{P}}$ that $|\text{supp}(\bar{\mathcal{P}}(\cdot|y_0))\cap\{\emptyset,\mathcal{S}^{-1}(\{y_0\})^c|=|\text{supp}(\widetilde{\mathcal{P}}(\cdot|y_0))\cap\{\emptyset,\mathcal{S}^{-1}(\{y_0\})^c|-1$ and $|\text{supp}(\bar{\mathcal{P}}(\cdot|y))|=|\text{supp}(\widetilde{\mathcal{P}}(\cdot|y))|$ for all other $y\in\Omega$.

Case 2: $y_0\in\mathcal{S}(R_0)$**.** Then letting

$$\bar{\mathcal{P}}(R|y)=\begin{cases}\widetilde{\mathcal{P}}(R_0|y)+\widetilde{\mathcal{P}}(R|y), & y=y_0, R=\mathcal{S}^{-1}(\{y_0\})\\ 0, & y=y_0, R=R_0\\ \widetilde{\mathcal{P}}(R|y), & \text{o.w.}\end{cases},$$

we observe that

$$\sum_{y\in\Omega}\sum_{R\in 2^{\Omega}} \rho(y)\widetilde{\mathcal{P}}(R|y)\mathbb{1}(y\in\mathcal{S}(R)) = \sum_{y\in\Omega}\sum_{R\in 2^{\Omega}} \rho(y)\bar{\mathcal{P}}(R|y)\mathbb{1}(y\in\mathcal{S}(R))$$

and $\bar{\mathcal{P}}$ satisfies all the constraints in Eq. (C.11) due to $\mathbb{1}(y \in \mathcal{S}(R_0)) \geq \mathbb{1}(y \in \mathcal{S}(\{y_0\}))$ for any $y \in \Omega$. From the construction of $\bar{\mathcal{P}}$, we know that $|\text{supp}(\bar{\mathcal{P}}(\cdot|y_0)) \cap \{\emptyset, \mathcal{S}^{-1}(\{y_0\})^c\}| = |\text{supp}(\widetilde{\mathcal{P}}(\cdot|y_0)) \cap \{\emptyset, \mathcal{S}^{-1}(\{y_0\})^c\}| - 1$ and $|\text{supp}(\bar{\mathcal{P}}(\cdot|y))| = |\text{supp}(\widetilde{\mathcal{P}}(\cdot|y))|$ for all other $y \in \Omega$.

Combining the above cases, we established our claim.

Finally, letting $\mathcal{P}^*(\cdot|y) = x^*(y) \cdot \delta_{\mathcal{S}^{-1}(\{y\})} + (1 - x^*(y)) \cdot \delta_{\emptyset}$ for all $y \in \Omega$, where x^* is the solution of Eq. (4.2), achieves the optimum value in Eq. (4.2). □

Appendix D

Appendix for Semantic-Aware Statistical Watermarking: SEAL

D.1 Proofs of SEAL Guarantees

D.1.1 Proof of Theorem 5.2.1

In the rest of this appendix, we will use $\rho_j(\cdot)$ and $q_j(\cdot)$ to abbreviate $\rho_j(\cdot|p, t_{1:j-1})$ and $q_j(\cdot|t_{j-K:j-1})$ respectively. For statistical analysis, we will also assume the pseudo-randomness functions used in Algorithm 5.2.1-5.2.2 are true random oracles. Our statistical results can be transformed into cryptography results by hardness hypothesis on the pseudo-random functions.

Proof. Fix h_j, the prompt p, and the history $t_{1:j-1}$. Let

$$\mu_{S,j} := h_j \sharp q_j, \qquad \mu_{\rho,j} := h_j \sharp \rho_j.$$

Algorithm 5.2.1 first samples $s_j \sim \mu_{S,j}$ and then constructs a code c_j by maximal coupling so that $c_j \sim \mu_{\rho,j}$ and

$$\Pr(c_j = s_j \mid p, t_{1:j-1}, h_j) = \sum_{c \in \Omega_h} \min\{\mu_{S,j}(c), \mu_{\rho,j}(c)\}.$$

When $b_{\text{logit}} = 0$, the final token is sampled from ρ_j conditional on $h_j(t_j) = c_j$. Therefore, for any token u,

$$\begin{aligned}
\mathbb{P}_{\text{SEAL}}(t_j = u \mid p, t_{1:j-1}, h_j) &= \sum_{c \in \Omega_h} \mathbb{P}_{\text{SEAL}}(c_j = c \mid p, t_{1:j-1}, h_j)\, \rho_j(u \mid h_j(u) = c, p, t_{1:j-1}) \\
&= \sum_{c \in \Omega_h} (h_j \sharp \rho_j)(c \mid p, t_{1:j-1})\, \rho_j(u \mid h_j(u) = c, p, t_{1:j-1}) \\
&= \rho_j(u \mid p, t_{1:j-1}). \qquad \text{(D.1)}
\end{aligned}$$

Averaging over the random hash function h_j gives the claimed distortion-free marginal. □

D.1.2 Proof of Theorem 5.2.2

Proof. Fix a non-watermarked candidate sequence y and a candidate window $I = [i, i + L]$. Conditional on y and on the reconstructed proposal distributions, the detector's randomness comes from the secret-keyed hash functions and seeds. Under H_0, the reconstructed seed s_j is independent of y_j and has distribution $h_j \sharp q_j$. Hence

$$\mathbb{P}(\xi_j = 1 \mid y) = \mathbb{P}\{h_j(y_j) = s_j \mid y\} = (h_j \sharp q_j)(h_j(y_j)) =: w_j.$$

If the threshold is computed using Eq. (5.4), then since

$$\mathbb{P}(Z_i = 1) = w_i,\ \mathbb{P}(Z_i = 0) = 1 - w_i,$$

where $w_j = h_j \sharp q_j(h_j(y_j)) = \mathbb{P}(Z_j = 1)$, we have

$$\begin{aligned}\mathbb{P}\left(\sum_{j=i}^{i+k} Z_j = l\right) &= \mathbb{P}(Z_{i+k} = 1) \cdot \mathbb{P}\left(\sum_{j=i}^{i+k-1} Z_j = l - 1\right) + \mathbb{P}(Z_{i+k} = 0) \cdot \mathbb{P}\left(\sum_{j=i}^{i+k-1} Z_j = l\right) \\ &= w_{i+k} \cdot \mathbb{P}\left(\sum_{j=i}^{i+k-1} Z_j = l - 1\right) + (1 - w_{i+k}) \cdot \mathbb{P}\left(\sum_{j=i}^{i+k-1} Z_j = l\right)\end{aligned}$$

It follows by induction that the $p_{k,l}$'s computed by Eq. (5.4) satisfy $p_{k,l} = \mathbb{P}\left(\sum_{j=i}^{i+k-1} Z_j = l\right)$. Therefore $\mathbb{P}_{\text{SEAL}}(\texttt{watermarked} = \text{True}) \leq \alpha$ holds by definition.

Next, we consider the case of using Eq. (5.5). Define

$$\mu = \sum_{j=i}^{i+L} h\#q_j(h_j(y_j)).$$

By Lemma D.1.1 and choice of ϵ, we have

$$\begin{aligned}\mathbb{P}\left(\sum_{j=i}^{i+L} \xi_j \geq (1+\epsilon)\mu\right) &\leq e^{(\epsilon - (1+\epsilon)\log(1+\epsilon))\mu} \\ &\leq \alpha.\end{aligned}$$

It follows that

$$\begin{aligned}\mathbb{P}_{\text{SEAL}}(\texttt{watermarked} = \text{True}) &= \mathbb{P}\left(\sum_{j=i}^{i+L} \xi_j \geq (1+\epsilon)\mu\right) \\ &\leq \alpha.\end{aligned}$$

□

Lemma D.1.1. *Let X be the sum of independent Bernoulli random variables (not necessarily with the same mean). Let $\mu = \mathbb{E}[X]$. Then for all $\epsilon > 0$,*

$$\mathbb{P}(X \geq (1+\epsilon)\mu) \leq e^{(\epsilon - (1+\epsilon)\log(1+\epsilon))\mu}.$$

Proof. Letting $X = \sum_{i=1}^{n} X_i$ where $X_i \sim \text{Bernoulli}(p_i)$. By a Chernoff bound,

$$\begin{aligned}\mathbb{P}(X \geq (1+\epsilon)\mu) &\leq \frac{\mathbb{E}\left[e^{\lambda X}\right]}{e^{(1+\epsilon)\mu\lambda}} \\ &= \frac{\prod_{i=1}^{n} \mathbb{E}\left[e^{\lambda X_i}\right]}{e^{(1+\epsilon)\mu\lambda}}\end{aligned}$$

Using the moment-generating function of the Bernoulli distribution,

$$\begin{aligned}\frac{\prod_{i=1}^{n} \mathbb{E}\left[e^{\lambda X_i}\right]}{e^{(1+\epsilon)\mu\lambda}} &= e^{\sum_{i=1}^{n} \log(p_i e^{\lambda} + 1 - p_i) - (1+\epsilon)\mu\lambda} \\ &\leq e^{\sum_{i=1}^{n} (p_i e^{\lambda} - p_i) - (1+\epsilon)\mu\lambda} \\ &= e^{\left(e^{\lambda} - 1 - (1+\epsilon)\lambda\right)\mu}.\end{aligned}$$

Since $e^{\lambda} - 1 - (1+\epsilon)\lambda \leq \epsilon - (1+\epsilon)\log(1+\epsilon)$ where the maximum is achieved at $\lambda^* = \log(1+\epsilon)$, we have

$$\mathbb{P}(X \geq (1+\epsilon)\mu) \leq e^{(\epsilon - (1+\epsilon)\log(1+\epsilon))\mu}.$$

□

Appendix E

Appendix for Anytime-Valid Statistical Watermarking

Notations. Let $[n]$ be the set $\{1, \ldots, n\}$ and $\Delta([n])$ denote the probability simplex over $[n]$. For the simplicity of notations, we let $\mathcal{V} = [n]$. Let S_n denote the permutation group of $\{1, \ldots, n\}$. For any $\sigma \in S_n$, σ^i means applying permutation σ i-times. For any matrix $M \in \mathbb{R}^{n \times n}$, let $M(x, y), M(x, :), M(:, y)$ denote the entry on x-th row and y-th column, the x-th row, and the y-th column respectively.

E.1 Proof of Theorem 6.4.1

Proof. Let

$$\mathcal{R} = \left\{ r \in \mathbb{R}^{n \times n} : \sum_{s \in \mathcal{V}} r(v, s) = 1,\ r(v, s) \geq 0,\ \forall v, s \in \mathcal{V} \right\}.$$

By Lemma E.1.7, the original problem is equivalent to

$$J' : \sup_{r \in \mathcal{R}} \inf_{q \in \mathcal{Q}(p_0, \delta)} \sup_{w \in \mathcal{P}(p_0, q)} \sum_{v,s \in \mathcal{V}} w(v, s) \cdot (\log r(v, s) - \log p_0(s)),$$

with $r(v, s) = p_0(s)e(v, s) \in \mathcal{R}$. Note that

$$\sum_{v,s \in \mathcal{V}} w(v, s) \cdot (-\log p_0(s)) = \sum_{s \in \mathcal{V}} p_0(s) \cdot (-\log p_0(s)) = H(p_0)$$

where $H(p_0)$ is the entropy of p_0. By Proposition E.1.6, the optimum value of J' is equal to

$$(1 - \tfrac{\delta}{2}) \log \left(1 - \frac{\delta}{2} \right) + \tfrac{\delta}{2} \log \left(\frac{\delta}{2(n-1)} \right),$$

achieved at

$$r^*(v,s) = \begin{cases} 1-\delta/2, & s = v, \\ \delta/(2n-2), & s \neq v, \end{cases}$$

and for any $q \in \mathcal{Q}(p_0, \delta)$ written as $q = \sum_{i=1}^{k} \lambda_i \cdot q_i$ for $q_i = p_0 + \frac{\delta}{2} \cdot (\mathbf{e}_{v_i} - \mathbf{e}_{s_i})$, the optimizer of the inner problem is given by

$$\sum_{i=1}^{k} \lambda_i \cdot \quad \frac{\delta}{2} \cdot (\mathbf{e}_{v_i}\mathbf{e}_{s_i}^\top - \mathbf{e}_{s_i}\mathbf{e}_{s_i}^\top) + \mathrm{diag}(p_0) \Bigg).$$

Therefore, the optimum value of the original problem is equal to

$$(1-\tfrac{\delta}{2}) \log \quad 1 - \frac{\delta}{2} \Bigg) + \tfrac{\delta}{2} \log \quad \frac{\delta}{2(n-1)} \Bigg) + H(p_0).$$

In particular, this is achieved at

$$e^*(v,s) = \begin{cases} \frac{1-\delta/2}{p_0(s)}, & s = v, \\ \frac{\delta}{2(n-1)p_0(s)}, & s \neq v, \end{cases}$$

For this e-value, the boundary couplings above attain the worst-case value, and interior targets are handled by maximal couplings as stated in Theorem 6.4.1. This completes the proof. □

E.1.1 Supporting lemma

Lemma E.1.1 (Diagonal Dominance)**.** *Let $\mathcal{V} = \{1, \ldots, n\}$. For any matrix $M_0 \in \mathbb{R}^{n \times n}$, there exists a permutation matrix $P \in \mathbb{R}^{n \times n}$ such that the following condition holds for any $v \in \mathcal{V}$ and permutation $\sigma \in S_n$:*

$$\sum_{i=0}^{\kappa_v - 1} (M_0 P)(\sigma^i(v), \sigma^i(v)) \geq \sum_{i=0}^{\kappa_v - 1} (M_0 P)(\sigma^i(v), \sigma^{i+1}(v)). \tag{E.1}$$

where κ_v is the minimum positive integer such that $\sigma^{\kappa_v}(v) = v$.

Proof. Let $\mathcal{P}$ be the set of all permutation matrices and

$$P^* = \arg\sup_{P \in \mathcal{P}} \mathrm{tr}[M_0 P].$$

Then we claim that P^* is the permutation matrix s.t. $M = M_0 P^*$ satisfying Eq. (E.1). Denote

$$C = \{v, \sigma(v), \ldots, \sigma^{\kappa_v - 1}(v)\}.$$

Suppose for the sake of contradiction that there exists $\sigma(\cdot)$ and v such that

$$\sum_{i=0}^{\kappa_v-1} M(\sigma^i(v), \sigma^i(v)) < \sum_{i=0}^{\kappa_v-1} M(\sigma^i(v), \sigma^{i+1}(v)). \tag{E.2}$$

Define permutation

$$\bar{\sigma}(u) = \begin{cases} \sigma^{-1}(u), & u \in C, \\ u, & u \notin C. \end{cases}$$

and permutation matrix

$$P_{\bar{\sigma}} = \begin{pmatrix} \mathbf{e}_{\bar{\sigma}(1)} & \cdots \mathbf{e}_{\bar{\sigma}(n)} \end{pmatrix}^\top$$

Then we have

$$\begin{aligned} MP_{\bar{\sigma}} &= \sum_{i=1}^{n} M(:,i) \cdot \mathbf{e}_{\bar{\sigma}(i)}^\top \\ &= \sum_{i \notin C} M(:,i) \cdot \mathbf{e}_{\bar{\sigma}(i)}^\top + \sum_{i \in C} M(:,i) \cdot \mathbf{e}_{\bar{\sigma}(i)}^\top \\ &= M + \sum_{i \in C} M(:,i) \cdot (\mathbf{e}_{\bar{\sigma}(i)}^\top - \mathbf{e}_i^\top) \\ &= M + \sum_{j=0}^{\kappa_v-1} M(:, \bar{\sigma}^j(v)) \cdot (\mathbf{e}_{\bar{\sigma}^{j+1}(v)}^\top - \mathbf{e}_{\bar{\sigma}^j(v)}^\top). \end{aligned}$$

Notice

$$\begin{aligned} \mathrm{tr}[P_{\bar{\sigma}} M] &= \mathrm{tr}[M] + \mathrm{tr}[\sum_{j=0}^{\kappa_v-1} M(:, \bar{\sigma}^j(v)) \cdot (\mathbf{e}_{\bar{\sigma}^{j+1}(v)}^\top - \mathbf{e}_{\bar{\sigma}^j(v)}^\top)] \\ &= \mathrm{tr}[M] + \mathrm{tr}[\sum_{j=0}^{\kappa_v-1} M(:, \bar{\sigma}^j(v)) \cdot \mathbf{e}_{\bar{\sigma}^{j+1}(v)}^\top] - \mathrm{tr}[\sum_{j=0}^{\kappa_v-1} M(:, \bar{\sigma}^j(v)) \cdot \mathbf{e}_{\bar{\sigma}^j(v)}^\top] \\ &= \mathrm{tr}[M] + \sum_{j=0}^{\kappa_v-1} M(\bar{\sigma}^{j+1}(v), \bar{\sigma}^j(v)) - \sum_{j=0}^{\kappa_v-1} M(\bar{\sigma}^j(v), \bar{\sigma}^j(v)) \\ &= \mathrm{tr}[M] + \sum_{j=0}^{\kappa_v-1} M(\sigma^j(v), \sigma^{j+1}(v)) - \sum_{j=0}^{\kappa_v-1} M(\sigma^j(v), \sigma^j(v)) \\ &> \mathrm{tr}[M] \end{aligned}$$

where the last inequality follows from Eq. (E.2).

But $\mathcal{P}$ is a group so $P^* P_{\bar{\sigma}} \in \mathcal{P}$. This contradicts with the definition of P^*. □

Lemma E.1.2 (Optimizer of Inner Problem)**.** *Let $\mathcal{V}$ be the set $\{1, \dots, n\}$ and $p_0 \in \Delta(\mathcal{V})$ be a distribution over $\mathcal{V}$ such that $\inf_{v\in\mathcal{V}} p_0(v) > \delta$. Let $M : \mathcal{V} \times \mathcal{V} \mapsto \mathbb{R}$ be a matrix that satisfies*

$$\sum_{i=0}^{\kappa_v - 1} M(\sigma^i(v), \sigma^i(v)) \geq \sum_{i=0}^{\kappa_v - 1} M(\sigma^i(v), \sigma^{i+1}(v)) \tag{E.3}$$

for any $v \in \mathcal{V}$, $\sigma \in S_n$, where $\kappa_v = \inf\{k \geq 1 : \sigma^k(v) = v\}$. Fix distinct $a, b \in \mathcal{V}$ and define

$$q = p_0 + \frac{\delta}{2} \cdot (\mathbf{e}_a - \mathbf{e}_b).$$

Consider the optimization problem

$$\sup_{w\in\mathcal{P}(p_0,q)} J(w), \qquad J(w) := \sum_{v,s\in\mathcal{V}} w(v,s)M(v,s),$$

where

$$\mathcal{P}(p_0, q) = \left\{ w : \mathcal{V} \times \mathcal{V} \to \mathbb{R}_+ \ \sum_{s\in\mathcal{V}} w(v,s) = q(v) \ \forall v \in \mathcal{V}, \ \sum_{v\in\mathcal{V}} w(v,s) = p_0(s) \ \forall s \in \mathcal{V} \right\}.$$

Then there exists a permutation $\sigma \in S_n$ such that

$$w^* = \frac{\delta}{2} \sum_{i=0}^{\kappa-2} \left(\mathbf{e}_{\sigma^i(a)} \mathbf{e}_{\sigma^{i+1}(a)}^\top - \mathbf{e}_{\sigma^{i+1}(a)} \mathbf{e}_{\sigma^{i+1}(a)}^\top \right) + \operatorname{diag}(p_0) \tag{E.4}$$

is an optimizer of J, where κ is the minimum positive integer such that $\sigma^\kappa(a) = a$.

Proof. First, we establish existence of an optimizer. Define w^{feas} by

$$\begin{aligned}
w^{\text{feas}}(a,b) &= \frac{\delta}{2}, \\
w^{\text{feas}}(v,v) &= \begin{cases} p_0(b) - \frac{\delta}{2}, & v = b, \\ p_0(v), & v \neq b, \end{cases} \\
w^{\text{feas}}(v,s) &= 0, \quad (v,s) \notin \{(a,b)\} \cup \{(v,v) : v \in \mathcal{V}\}.
\end{aligned}$$

Because $\inf_v p_0(v) > \delta > \delta/2$, we have $w^{\text{feas}} \geq 0$. Its row sums satisfy

$$\begin{aligned}
\sum_s w^{\text{feas}}(a,s) &= p_0(a) + \frac{\delta}{2} = q(a), \\
\sum_s w^{\text{feas}}(b,s) &= p_0(b) - \frac{\delta}{2} = q(b), \\
\sum_s w^{\text{feas}}(v,s) &= p_0(v) = q(v), \quad v \notin \{a,b\},
\end{aligned}$$

and its column sums satisfy

$$\sum_v w^{\text{feas}}(v,b) = p_0(b) - \frac{\delta}{2} + \frac{\delta}{2} = p_0(b),$$
$$\sum_v w^{\text{feas}}(v,s) = p_0(s), \quad s \neq b.$$

Thus $w^{\text{feas}} \in \mathcal{P}(p_0, q)$. The set $\mathcal{P}(p_0, q)$ is a nonempty bounded polytope, and J is linear, so there exists

$$w^{(0)} \in \arg \sup_{p \in \mathcal{P}(p_0,q)} J(w).$$

For a feasible w, let G_w be the directed graph on vertex set $\mathcal{V}$ with an edge (v, s) whenever $v \neq s$ and $w(v, s) > 0$. A directed cycle in G_w is a k-tuple $(v_0, \ldots, v_{k-1})$ of distinct vertices, $k \geq 2$, such that

$$w(v_i, v_{i+1}) > 0, \quad i = 0, \ldots, k-1,$$

with indices understood modulo k (so $v_k = v_0$).

Next, we show that for an optimal distribution $\bar{w}$, the graph $G_{\bar{w}}$ has no directed cycles. Fix any feasible w and such a cycle $(v_0, \ldots, v_{k-1})$. For $\varepsilon > 0$ define $\tilde{w}$ by

$$\tilde{w}(v_i, v_{i+1}) = w(v_i, v_{i+1}) - \varepsilon, \qquad i = 0, \ldots, k-1, \tag{E.5}$$
$$\tilde{w}(v_i, v_i) = w(v_i, v_i) + \varepsilon, \qquad i = 0, \ldots, k-1, \tag{E.6}$$
$$\tilde{w}(x, y) = w(x, y), \qquad (x, y) \notin \{(v_i, v_{i+1}), (v_i, v_i) : 0 \leq i \leq k-1\}.$$

Choose

$$0 < \varepsilon \leq \inf_{0 \leq i \leq k-1} w(v_i, v_{i+1}),$$

so that $\tilde{w} \geq 0$.

Row sums. For each i,

$$\begin{aligned}\sum_{s \in \mathcal{V}} \tilde{w}(v_i, s) &= (w(v_i, v_{i+1}) - \varepsilon) + (w(v_i, v_i) + \varepsilon) + \sum_{s \notin \{v_i, v_{i+1}\}} w(v_i, s) \\ &= \sum_{s \in \mathcal{V}} w(v_i, s).\end{aligned}$$

All other rows are unchanged, so

$$\sum_s \tilde{w}(v, s) = \sum_s w(v, s) = q(v), \qquad \forall v \in \mathcal{V}. \tag{E.7}$$

Column sums. For each i,

$$\begin{aligned}\sum_{x\in\mathcal{V}} \tilde{w}(x, v_i) =\ & (w(v_{i-1}, v_i) - \varepsilon) + (w(v_i, v_i) + \varepsilon) + \sum_{x\notin\{v_{i-1},v_i\}} w(x, v_i) \\ =\ & \sum_{x\in\mathcal{V}} w(x, v_i),\end{aligned}$$

where again indices are taken modulo k. All other columns are unchanged, so

$$\sum_v \tilde{w}(v, s) = \sum_v w(v, s) = p_0(s), \qquad \forall s \in \mathcal{V}. \tag{E.8}$$

Thus $\tilde{w} \in \mathcal{P}(p_0, q)$.

Objective value. Only entries on the cycle and the corresponding diagonals change, hence

$$\begin{aligned}J(\tilde{w}) - J(w) =\ & \sum_{i=0}^{k-1}\Big[\tilde{w}(v_i, v_i)M(v_i, v_i) + \tilde{w}(v_i, v_{i+1})M(v_i, v_{i+1}) \qquad \text{(E.9)} \\ & - w(v_i, v_i)M(v_i, v_i) - w(v_i, v_{i+1})M(v_i, v_{i+1})\Big] \\ =\ & \varepsilon \sum_{i=0}^{k-1} \left(M(v_i, v_i) - M(v_i, v_{i+1})\right). \qquad \text{(E.10)}\end{aligned}$$

Let $\sigma \in S_n$ be the permutation whose cycle on the set $\{v_0, \ldots, v_{k-1}\}$ is $(v_0 v_1 \ldots v_{k-1})$ and which fixes all other vertices. For $v = v_0$ we have $\kappa_v = k$ and

$$\sigma^i(v_0) = v_i, \quad \sigma^{i+1}(v_0) = v_{i+1}, \quad i = 0, \ldots, k-1,$$

so by Eq. (E.3),

$$\begin{aligned}\sum_{i=0}^{k-1} M(v_i, v_i) =\ & \sum_{i=0}^{\kappa_v-1} M(\sigma^i(v_0), \sigma^i(v_0)) \\ \geq\ & \sum_{i=0}^{\kappa_v-1} M(\sigma^i(v_0), \sigma^{i+1}(v_0)) = \sum_{i=0}^{k-1} M(v_i, v_{i+1}). \qquad \text{(E.11)}\end{aligned}$$

Combining Eq. (E.10) and Eq. (E.11) yields

$$J(\tilde{w}) - J(w) \geq 0. \tag{E.12}$$

Now start from the optimizer $w^{(0)}$. If $G_{w^{(0)}}$ has no directed cycles, set $\bar{w} = w^{(0)}$. Otherwise, choose a directed cycle in $G_{w^{(0)}}$, apply the above transformation with maximal ε as chosen above, and obtain $w^{(1)} \in \mathcal{P}(p_0, q)$ with $J(w^{(1)}) \geq J(w^{(0)})$. Since $w^{(0)}$ is optimal, $J(w^{(1)}) = J(w^{(0)})$, so $w^{(1)}$ is also optimal. Moreover, at least one edge of the chosen cycle has $w^{(1)}(v_i, v_{i+1}) = 0$.

Iterating this construction, we obtain a sequence of optimal plans $w^{(m)}$ in $\mathcal{P}(p_0, q)$ in which the set of off-diagonal edges with positive mass strictly decreases whenever there is a directed cycle. Since there are only finitely many off-diagonal entries, this procedure must terminate. We thus obtain an optimal plan $\bar{w}$ such that $G_{\bar{w}}$ contains no directed cycle.

Define the off-diagonal flow

$$F(v, s) := \begin{cases} \bar{w}(v, s), & v \neq s, \\ 0, & v = s. \end{cases}$$

For $v \in \mathcal{V}$ define

$$\mathrm{out}(v) := \sum_{s \neq v} F(v, s), \qquad \mathrm{in}(v) := \sum_{u \neq v} F(u, v).$$

Using the marginal constraints of $\bar{w}$,

$$\begin{aligned} q(v) &= \sum_s \bar{w}(v, s) = \bar{w}(v, v) + \mathrm{out}(v), \\ p_0(v) &= \sum_u \bar{w}(u, v) = \bar{w}(v, v) + \mathrm{in}(v), \end{aligned}$$

so

$$\mathrm{out}(v) - \mathrm{in}(v) = q(v) - p_0(v) = \begin{cases} \frac{\delta}{2}, & v = a, \\ -\frac{\delta}{2}, & v = b, \\ 0, & v \notin \{a, b\}. \end{cases} \tag{E.13}$$

Thus F is a nonnegative flow of value $\delta/2$ from a to b on the directed acyclic graph $G_{\bar{w}}$.

We now decompose F into simple a–b paths. Define $F^{(0)} := F$ and proceed inductively. Assume $F^{(m)}$ is a nonnegative flow from a to b with balance equation Eq. (E.13). Since $\mathrm{out}(a) - \mathrm{in}(a) = \delta/2 > 0$, there exists $s_1 \neq a$ with $F^{(m)}(a, s_1) > 0$. Set $v_0^{(m)} := a$ and $v_1^{(m)} := s_1$.

Suppose $v_0^{(m)} = a, \ldots, v_j^{(m)} \neq b$ are already recursively constructed such that we use Eq. (E.13) and nonnegativity to obtain for $j \geq 1$,

$$\mathrm{in}(v_j^{(m)}) \geq F^{(m)}(v_{j-1}^{(m)}, v_j^{(m)}) > 0.$$

For $v_j^{(m)} \notin \{a, b\}$, Eq. (E.13) gives $\mathrm{out}(v_j^{(m)}) = \mathrm{in}(v_j^{(m)}) > 0$, so there exists $v_{j+1}^{(m)} \neq v_j^{(m)}$ with $F^{(m)}(v_j^{(m)}, v_{j+1}^{(m)}) > 0$. Since $G_{\bar{w}}$ is acyclic and finite, the sequence $v_0^{(m)}, v_1^{(m)}, \ldots$ cannot visit a vertex twice; hence the process must terminate at a vertex with no outgoing edges. By (E.13), such a vertex must satisfy $\mathrm{out}(v) - \mathrm{in}(v) \leq 0$. Because all intermediate vertices have balance 0, the only possible terminal vertex is b. Thus we obtain a simple path

$$P^{m+1} : a = v_0^{(m)} \to v_1^{(m)} \to \cdots \to v_{k_m}^{(m)} = b.$$

Let

$$\alpha_m := \inf_{0\le i\le k_m-1} F^{(m)}(v_i^{(m)}, v_{i+1}^{(m)}) > 0,$$

and define

$$F^{(m+1)}(v,s) := F^{(m)}(v,s) - \alpha_m \cdot \mathbf{1}\{(v,s) = (v_i^{(m)}, v_{i+1}^{(m)}) \text{ for some } i = 0,\dots,k_m-1\}.$$

Then $F^{(m+1)} \ge 0$ and satisfies the same balance equations Eq. (E.13), but has strictly smaller total flow $\sum_{v,s} F^{(m+1)}(v,s) = \sum_{v,s} F^{(m)}(v,s) - \alpha_m$.

Since the total flow is initially $\delta/2$ and decreases by a positive amount at each step, this procedure terminates after some L steps with $F^{(L)} \equiv 0$. For $\ell = 1,\dots,L$, we obtain simple paths

$$P^\ell : a = v_0^\ell \to v_1^\ell \to \cdots \to v_{k_\ell}^\ell = b, \qquad \ell = 1,\dots,L,$$

and coefficients $\alpha_\ell > 0$ such that

$$F(v,s) = \sum_{\ell=0}^{L-1} \alpha_\ell \mathbf{1}\{(v,s) = (v_i^\ell, v_{i+1}^\ell) \text{ for some } i = 0,\dots,k_\ell-1\}, \qquad v \ne s, \tag{E.14}$$

$$\sum_{\ell=0}^{L-1} \alpha_\ell = \frac{\delta}{2}. \tag{E.15}$$

Finally, we prove that the optimal G_{w^*} contains only a single path and derive the closed form solution w^*.

For a general feasible $w \in \mathcal{P}(p_0, q)$, the column constraints give

$$w(v,v) = p_0(v) - \sum_{u\ne v} w(u,v), \qquad v \in \mathcal{V}. \tag{E.16}$$

Thus

$$\begin{aligned} J(w) &= \sum_v w(v,v)M(v,v) + \sum_{u\ne v} w(u,v)M(u,v) \\ &= \sum_v \left(p_0(v) - \sum_{u\ne v} w(u,v)\right) M(v,v) + \sum_{u\ne v} w(u,v)M(u,v) \\ &= \underbrace{\sum_v p_0(v)M(v,v)}_{=:J_0} + \sum_{u\ne v} w(u,v)\left(M(u,v) - M(v,v)\right). \end{aligned} \tag{E.17}$$

For the optimal plan $\bar{w}$, the off-diagonal part is F, so

$$J(\bar{w}) = J_0 + \sum_{u\ne v} F(u,v)\left(M(u,v) - M(v,v)\right). \tag{E.18}$$

For each path P^ℓ, define its *gain*

$$W(P^\ell) := \sum_{i=0}^{k_\ell - 1} \Big(M(v_i^\ell, v_{i+1}^\ell) - M(v_{i+1}^\ell, v_{i+1}^\ell) \Big). \tag{E.19}$$

Using the decomposition Eq. (E.14), we obtain from Eq. (E.18)

$$\begin{aligned} J(\bar{w}) &= J_0 + \sum_{u \neq v} \left(\sum_{\ell=1}^{L} \alpha_\ell \mathbf{1}\{(u,v) = (v_i^\ell, v_{i+1}^\ell) \text{ for some } i = 0, ..., k_\ell - 1\} \right) (M(u,v) - M(v,v)) \\ &= J_0 + \sum_{\ell=1}^{L} \alpha_\ell \sum_{i=0}^{k_\ell - 1} \Big(M(v_i^\ell, v_{i+1}^\ell) - M(v_{i+1}^\ell, v_{i+1}^\ell) \Big) \\ &= J_0 + \sum_{\ell=1}^{L} \alpha_\ell W(P^\ell). \end{aligned} \tag{E.20}$$

By Eq. (E.15),

$$J(\bar{w}) = J_0 + \frac{\delta}{2} \sum_{\ell=1}^{L} \theta_\ell W(P^\ell), \qquad \theta_\ell := \frac{\alpha_\ell}{\delta/2}, \quad \theta_\ell \geq 0, \ \sum_\ell \theta_\ell = 1. \tag{E.21}$$

Let

$$W^* := \sup\{W(P) : P \text{ is a simple directed path from } a \text{ to } b\}. \tag{E.22}$$

Since the graph on $\mathcal{V}$ is finite, the maximum exists. From Eq. (E.21) we deduce

$$J(\bar{w}) \leq J_0 + \frac{\delta}{2} W^*. \tag{E.23}$$

Now fix an arbitrary simple directed path

$$P : a = u_0 \to u_1 \to \cdots \to u_K = b$$

with distinct vertices $u_0, \ldots, u_K$. Define w^P by

$$w^P := \frac{\delta}{2} \sum_{i=0}^{K-1} \Big(\mathbf{e}_{u_i} \mathbf{e}_{u_{i+1}}^\top - \mathbf{e}_{u_{i+1}} \mathbf{e}_{u_{i+1}}^\top \Big) + \operatorname{diag}(p_0). \tag{E.24}$$

We first verify $w^P \in \mathcal{P}(p_0, q)$.
The diagonal entries of w^P are

$$\begin{aligned} w^P(u_0, u_0) &= p_0(u_0), \\ w^P(u_j, u_j) &= p_0(u_j) - \frac{\delta}{2}, \quad 1 \leq j \leq K, \\ w^P(v, v) &= p_0(v), \quad v \notin \{u_0, \ldots, u_K\}. \end{aligned} \tag{E.25}$$

Since $\inf_v p_0(v) > \delta$, we have $p_0(u_j) - \delta/2 > \delta/2 > 0$ for $1 \le j \le K$; thus all diagonals are nonnegative. Off-diagonal entries are either 0 or $\delta/2$, so $w^P \ge 0$.

For $v = u_0 = a$,

$$\sum_s w^P(a,s) = w^P(a,a) + w^P(a,u_1) = p_0(a) + \frac{\delta}{2} = q(a).$$

For an internal vertex u_j with $1 \le j \le K-1$,

$$\begin{aligned}\sum_s w^P(u_j,s) &= w^P(u_j,u_j) + w^P(u_j,u_{j+1}) \\ &= \quad p_0(u_j) - \frac{\delta}{2}\Big) + \frac{\delta}{2} = p_0(u_j) = q(u_j).\end{aligned}$$

For $v = u_K = b$,

$$\sum_s w^P(b,s) = w^P(b,b) = p_0(b) - \frac{\delta}{2} = q(b).$$

For $v \notin \{u_0, \dots, u_K\}$, the only nonzero entry in row v is the diagonal, so

$$\sum_s w^P(v,s) = p_0(v) = q(v).$$

Thus the row constraints are satisfied.

For a vertex u_{i+1} on the path (with $0 \le i \le K-1$),

$$\begin{aligned}\sum_v w^P(v,u_{i+1}) &= w^P(u_{i+1},u_{i+1}) + w^P(u_i,u_{i+1}) \\ &= \quad p_0(u_{i+1}) - \frac{\delta}{2}\Big) + \frac{\delta}{2} = p_0(u_{i+1}).\end{aligned}$$

For $s \notin \{u_1, \dots, u_K\}$, the only nonzero entry in column s is $w^P(s,s) = p_0(s)$, so

$$\sum_v w^P(v,s) = p_0(s).$$

Hence $w^P \in \mathcal{P}(p_0, q)$.

From Eq. (E.24) and linearity of J,

$$J(w^P) - J_0 = \frac{\delta}{2} \sum_{i=0}^{K-1} \left(M(u_i,u_{i+1}) - M(u_{i+1},u_{i+1})\right) = \frac{\delta}{2} W(P), \tag{E.26}$$

where $W(P)$ is defined as in Eq. (E.19) for this path P.

Now choose a path P^* attaining the maximum gain $W(P^*) = W^*$ in Eq. (E.22), and set $w^* := w^{P^*}$. Then

$$\begin{aligned} J(w^*) &= J_0 + \frac{\delta}{2} W^* \geq J_0 + \frac{\delta}{2} W(P^\ell) \quad \forall \ell, \\ &\geq J_0 + \frac{\delta}{2} \sum_{\ell=1}^{L} \theta_\ell W(P^\ell) = J(\bar{w}) \geq J(w), \quad \forall p \in \mathcal{P}(p_0, q), \end{aligned} \tag{E.27}$$

where we used Eq. (E.21) and Eq. (E.23) in the last line. Thus, w^* is an optimizer of J over $\mathcal{P}(p_0, q)$ with only a single path.

Write the maximizing path as

$$P^* : a = u_0 \to u_1 \to \cdots \to u_K = b.$$

Define a permutation $\sigma \in S_n$ by

$$\begin{aligned} \sigma(u_i) &= u_{i+1}, \quad i = 0, \ldots, K-1, \\ \sigma(u_K) &= u_0, \\ \sigma(v) &= v, \quad v \notin \{u_0, \ldots, u_K\}. \end{aligned}$$

Then the orbit of a under σ is

$$a, \sigma(a), \ldots, \sigma^K(a) = u_0, u_1, \ldots, u_K,$$

and $\sigma^{K+1}(a) = a$, so the minimal positive integer κ with $\sigma^\kappa(a) = a$ is $\kappa = K+1$. In particular,

$$\sigma^i(a) = u_i, \quad i = 0, \ldots, K.$$

Therefore, the definition Eq. (E.24) of w^* can be rewritten as

$$\begin{aligned} w^* &= \frac{\delta}{2} \sum_{i=0}^{K-1} \left(\mathbf{e}_{u_i} \mathbf{e}_{u_{i+1}}^\top - \mathbf{e}_{u_{i+1}} \mathbf{e}_{u_{i+1}}^\top \right) + \operatorname{diag}(p_0) \\ &= \frac{\delta}{2} \sum_{i=0}^{\kappa-2} \left(\mathbf{e}_{\sigma^i(a)} \mathbf{e}_{\sigma^{i+1}(a)}^\top - \mathbf{e}_{\sigma^{i+1}(a)} \mathbf{e}_{\sigma^{i+1}(a)}^\top \right) + \operatorname{diag}(p_0), \end{aligned}$$

which is exactly Eq. (E.4). This completes the proof. □

Lemma E.1.3 (Optimizer of Middle Problem)**.** *Let $\mathcal{V} = \{1, \ldots, n\}$, $p_0 \in \Delta(\mathcal{V})$ be a distribution over $\mathcal{V}$ such that $\inf_{v \in \mathcal{V}} p_0(v) > \delta$, and $M : \mathcal{V} \times \mathcal{V} \mapsto \mathbb{R}$ be a matrix. Consider the optimization problem*

$$J : \inf_{q \in \mathcal{Q}(p_0, \delta)} \sup_{w \in \mathcal{P}(p_0, q)} \sum_{v,s \in \mathcal{V}} w(v, s) \cdot M(v, s)$$

where

$$\begin{aligned}\mathcal{Q}(p_0,\delta) =& \ \{q \in \Delta(\mathcal{V}) : \|q - p_0\|_1 \le \delta\} \\ \mathcal{P}(p_0,q) =& \ \left\{w(v,s) : \sum_s w(v,s) = q(v), \sum_v w(v,s) = p_0(s)\right\}\end{aligned}$$

Then there exists an optimal solution q^* of J coming from the set

$$\mathcal{Q}_{\text{ext}}(p_0,\delta) := \left\{p_0 + \frac{\delta}{2}\cdot(\mathbf{e}_a - \mathbf{e}_b) : a \neq b \in \mathcal{V}\right\}.$$

Proof. By Lemma E.3.1, $\mathcal{Q}_{\text{ext}}(p_0,\delta)$ is the set of extreme points of the convex polytope $\mathcal{Q}(p_0,\delta)$. Define $J(q) = \sup_{w\in\mathcal{P}(p_0,q)} \sum_{v,s\in\mathcal{V}} w(v,s)\cdot M(v,s)$. We show that $J(q)$ is a concave function. Indeed for all $q_1 \neq q_2$, we let

$$\begin{aligned}w_1 :=& \ \arg\sup_{w\in\mathcal{P}(p_0,q_1)} \sum_{v,s\in\mathcal{V}} w(v,s)\cdot M(v,s) \\ w_2 :=& \ \arg\sup_{w\in\mathcal{P}(p_0,q_2)} \sum_{v,s\in\mathcal{V}} w(v,s)\cdot M(v,s).\end{aligned}$$

We have

$$\begin{aligned}J((q_1+q_2)/2) =& \sup_{w\in\mathcal{P}(p_0,(q_1+q_2)/2)} \sum_{v,s\in\mathcal{V}} w(v,s)\cdot M(v,s) \\ \ge& \sum_{v,s\in\mathcal{V}} (w_1(v,s)+w_2(v,s))/2\cdot M(v,s) \\ =& \ (J(q_1)+J(q_2))/2,\end{aligned}$$

where the second step is because the feasibility of $(w_1(v,s)+w_2(v,s))/2$:

$$\sum_s (w_1(v,s)+w_2(v,s))/2 = (q_1(v)+q_2(v))/2.$$

Since $J(q)$ is concave over a convex feasible set, its optimizer must be found on the extreme set $\mathcal{Q}_{\text{ext}}(p_0,\delta)$. This completes the proof. □

Lemma E.1.4 (Optimal Subpaths)**.** *Let $\mathcal{V} = \{1,\dots,n\}$, $p_0 \in \Delta(\mathcal{V})$ be a distribution over $\mathcal{V}$ such that $\inf_{v\in\mathcal{V}} p_0(v) > \delta$. Let $M : \mathcal{V}\times\mathcal{V} \mapsto \mathbb{R}$ be a matrix that satisfies*

$$\sum_{i=0}^{\kappa_v - 1} M(\sigma^i(v),\sigma^i(v)) \ge \sum_{i=0}^{\kappa_v - 1} M(\sigma^i(v),\sigma^{i+1}(v))$$

for any $v \in \mathcal{V}$, $\sigma \in S_n$, where $\kappa_v = \inf\{k \ge 1 : \sigma^k(v) = v\}$. Consider the optimization problem

$$J(q) = \sup_{w\in\mathcal{P}(p_0,q)} \sum_{v,s\in\mathcal{V}} w(v,s)\cdot M(v,s)$$

where

$$\mathcal{P}(p_0, q) = \left\{ w(v,s) : \sum_s w(v,s) = q(v), \sum_v w(v,s) = p_0(s) \right\}.$$

If there exists $q = p_0 + \frac{\delta}{2} \cdot (\mathbf{e}_a - \mathbf{e}_b)$ *for some* $a \neq b \in \mathcal{V}$ *and* w_q^* *such that* w_q^* *is the optimizer of* $J(q)$ *and for a permutation* σ *we have that*

$$w^*(q) = \frac{\delta}{2} \cdot \sum_{i=0}^{\kappa-2} (\mathbf{e}_{\sigma^i(a)} \mathbf{e}_{\sigma^{i+1}(a)}^\top - \mathbf{e}_{\sigma^{i+1}(a)} \mathbf{e}_{\sigma^{i+1}(a)}^\top) + \operatorname{diag}(p_0),$$

where κ *is the minimum positive integer such that* $\sigma^\kappa(a) = a$*. Then define* $c := \sigma^i(a)$ *and* $d := \sigma^{i+1}(a)$ *with* $i < \kappa$*, and let* $q' = p_0 + \frac{\delta}{2} \cdot (\mathbf{e}_c - \mathbf{e}_d)$*. We have that the* $w^*(q')$ *that optimizes* $J(q')$ *is given by*

$$w^*(q') = \frac{\delta}{2} \cdot (\mathbf{e}_c \mathbf{e}_d^\top - \mathbf{e}_d \mathbf{e}_d^\top) + \operatorname{diag}(p_0).$$

Proof. Let $c = \sigma^i(a)$ and $d = \sigma^{i+1}(a)$. Define the local transport term corresponding to the edge (c, d) as:

$$T_{c,d} := \frac{\delta}{2} \cdot (\mathbf{e}_c \mathbf{e}_d^\top - \mathbf{e}_d \mathbf{e}_d^\top).$$

Note that the proposed optimizer for the subproblem $J(q')$ is given by $\hat{w} = \operatorname{diag}(p_0) + T_{c,d}$.

We proceed by contradiction. Assume that $\hat{w}$ is not the optimizer for $J(q')$. Then there exists a feasible transport plan $\tilde{w} \in \mathcal{P}(p_0, q')$ such that:

$$\sum_{v,s \in \mathcal{V}} \tilde{w}(v,s) M(v,s) > \sum_{v,s \in \mathcal{V}} \hat{w}(v,s) M(v,s).$$

By Lemma E.1.2, $\tilde{w}$ can be chosen to take the form of

$$\tilde{w} = \frac{\delta}{2} \sum_{i=0}^{\bar{\kappa}-2} \left(\mathbf{e}_{\bar{\sigma}^i(c)} \mathbf{e}_{\sigma^{i+1}(c)}^\top - \mathbf{e}_{\bar{\sigma}^{i+1}(c)} \mathbf{e}_{\bar{\sigma}^{i+1}(c)}^\top \right) + \operatorname{diag}(p_0)$$

for some $\bar{\sigma} \in S_n$ and $\bar{\kappa} = \inf\{k : \bar{\sigma}^k(c) = c\}$. It follows that $\inf_{v \in \mathcal{V}} (\tilde{w} - \operatorname{diag}(p_0))(v,v) \geq -\delta/2$ and $\inf_{v \neq s \in \mathcal{V}} (\tilde{w} - \operatorname{diag}(p_0))(v,s) \geq 0$.

Substituting $\hat{w} = \operatorname{diag}(p_0) + T_{c,d}$ into the inequality, we have:

$$\sum_{v,s \in \mathcal{V}} (\tilde{w}(v,s) - \operatorname{diag}(p_0)_{v,s}) M(v,s) > \sum_{v,s \in \mathcal{V}} (T_{c,d})_{v,s} M(v,s). \tag{E.28}$$

Now, consider the global optimizer $w^*(q)$. By the hypothesis, $w^*(q)$ decomposes into a sum of path segments. We can separate the specific term $T_{c,d}$ from the rest of the path:

$$w^*(q) = \underbrace{\left(\operatorname{diag}(p_0) + \frac{\delta}{2} \sum_{j \neq i}^{\kappa-2} (\mathbf{e}_{\sigma^j(a)} \mathbf{e}_{\sigma^{j+1}(a)}^\top - \mathbf{e}_{\sigma^{j+1}(a)} \mathbf{e}_{\sigma^{j+1}(a)}^\top) \right)}_{R} + T_{c,d},$$

where R represents the flow on the path excluding the step from c to d. We construct a new global transport plan w_{new} by replacing the local step $T_{c,d}$ in $w^*(q)$ with the "better" local flow derived from $\tilde{w}$:

$$w_{\text{new}} := R + (\tilde{w} - \text{diag}(p_0)).$$

Substituting $R = w^*(q) - T_{c,d}$, we get:

$$w_{\text{new}} = w^*(q) - T_{c,d} + \tilde{w} - \text{diag}(p_0).$$

We verify that w_{new} is feasible for the original problem $J(q)$:

Since $\tilde{w} \in \mathcal{P}(p_0, q')$, its row sum is $q' = p_0 + \frac{\delta}{2}(\mathbf{e}_c - \mathbf{e}_d)$. The term $T_{c,d}$ also corresponds to a row marginal shift of $\frac{\delta}{2}(\mathbf{e}_c - \mathbf{e}_d)$. Thus, w_{new} preserves the row sums of $w^*(q)$, which equal q.

Both $\tilde{w}$ and $\text{diag}(p_0) + T_{c,d}$ maintain column sums equal to p_0. Thus, w_{new} maintains the column sums of $w^*(q)$, which equal p_0.

Since $\inf_v p_0(v) > \delta$, $\inf_v T_{c,d}(v, v) \geq -\delta/2$, and $\inf_v (\tilde{w} - \text{diag}(p_0))(v, v) \geq -\delta/2$, we have $\inf_v w_{\text{new}}(v, v) \geq 0$. Furthermore, all off-diagonal entries of $w^*(q) - T_{c,d}$ and $\tilde{w} - \text{diag}(p_0)$ are non-negative, thus we conclude that w_{new} is non-negative.

Finally, we compare the objective value of w_{new} to $w^*(q)$:

$$\begin{aligned} J(w_{\text{new}}) &= \sum_{v,s} (R_{v,s} + \tilde{w}(v, s) - \text{diag}(p_0)_{v,s}) M(v, s) \\ &= \sum_{v,s} R_{v,s} M(v, s) + \sum_{v,s} (\tilde{w}(v, s) - \text{diag}(p_0)_{v,s}) M(v, s) \\ &> \sum_{v,s} R_{v,s} M(v, s) + \sum_{v,s} (T_{c,d})_{v,s} M(v, s) \quad \text{(by Eq. (E.28))} \\ &= J(w^*(q)). \end{aligned}$$

We have constructed a feasible solution $w_{\text{new}} \in \mathcal{P}(p_0, q)$ with a strictly higher objective value than $w^*(q)$. This contradicts the optimality of $w^*(q)$. Therefore, $\hat{w}$ must be the optimizer for $J(q')$. □

Corollary E.1.5 (Boundary reduction)**.** *Fix any nonnegative row-stochastic matrix r, and define*

$$\Phi_r(q) := \sup_{w \in \mathcal{P}(p_0, q)} \sum_{v,s \in \mathcal{V}} w(v, s) \log r(v, s).$$

Then

$$\inf_{q \in \mathcal{Q}(p_0, \delta)} \Phi_r(q) = \inf_{q \in \mathcal{Q}_{\text{ext}}(p_0, \delta)} \Phi_r(q),$$

where

$$\mathcal{Q}_{\text{ext}}(p_0, \delta) = \left\{ p_0 + \frac{\delta}{2}(\mathbf{e}_a - \mathbf{e}_b) : a \neq b \in \mathcal{V} \right\}.$$

Proof. For fixed r, the function $\Phi_r(q)$ is concave in q. Indeed, if $w_i \in \mathcal{P}(p_0, q_i)$ is optimal for q_i, then $\lambda w_1 + (1-\lambda)w_2 \in \mathcal{P}(p_0, \lambda q_1 + (1-\lambda)q_2)$, and therefore

$$\Phi_r(\lambda q_1 + (1-\lambda)q_2) \geq \lambda \Phi_r(q_1) + (1-\lambda)\Phi_r(q_2).$$

By Lemma E.3.1, the compact polytope $\mathcal{Q}(p_0, \delta)$ is the convex hull of $\mathcal{Q}_{\text{ext}}(p_0, \delta)$. If $q = \sum_i \lambda_i q_i$ is any convex representation by extreme points, then concavity gives

$$\Phi_r(q) \geq \sum_i \lambda_i \Phi_r(q_i) \geq \min_i \Phi_r(q_i).$$

Thus the minimum over $\mathcal{Q}(p_0, \delta)$ is attained at an extreme point, which proves the claim. □

Proposition E.1.6 (Reformulated Problem)**.** *Let $\mathcal{V} = \{1, \ldots, n\}$ and $p_0 \in \Delta(\mathcal{V})$ be a distribution over $\mathcal{V}$ such that $\inf_{v\in\mathcal{V}} p_0(v) > \delta$. Consider the following problem*

$$\sup_r \inf_{q\in\mathcal{Q}(p_0,\delta)} \sup_{w\in\mathcal{P}(p_0,q)} \sum_{v,s\in\mathcal{V}} w(v,s) \cdot \log r(v,s) \tag{E.29}$$

$$s.t. \sum_{s\in\mathcal{V}} r(v,s) = 1, \ \forall v \in \mathcal{V} \tag{E.30}$$

$$r(v,s) \geq 0, \ \forall v, s \in \mathcal{V}$$

where p_0 is a fixed distribution over the sample space $\mathcal{V}$ such that $\inf_{v\in\mathcal{V}} p_0(v) > \delta$, and

$$\mathcal{Q}(p_0, \delta) = \{q \in \Delta(\mathcal{V}) : \|q - p_0\|_1 \leq \delta\}$$
$$\mathcal{P}(p_0, q) = \left\{ w(v,s) : \sum_{s\in\mathcal{V}} w(v,s) = q(v), \sum_{v\in\mathcal{V}} w(v,s) = p_0(s) \right\}.$$

Then the optimum value is

$$J^* = (1 - \tfrac{\delta}{2}) \log \left(1 - \frac{\delta}{2}\right) + \tfrac{\delta}{2} \log \left(\frac{\delta}{2(n-1)}\right).$$

In particular, this is achieved at

$$r^*_{a,b}(v,s) = \begin{cases} 1 - \delta/2, & s = v, \\ \delta/(2n-2), & s \neq v, \end{cases}$$

and for every boundary point $q_{a,b} = p_0 + \frac{\delta}{2}(\mathbf{e}_a - \mathbf{e}_b)$, $a \neq b$, the inner problem is optimized by

$$w^*_{a,b} = \operatorname{diag}(p_0) + \frac{\delta}{2}\left(\mathbf{e}_a \mathbf{e}_b^\top - \mathbf{e}_b \mathbf{e}_b^\top\right).$$

For a general $q \in \mathcal{Q}(p_0, \delta)$, the inner optimizer for the proposed r^* is any maximal coupling of q and p_0. Since the objective places larger weight on diagonal entries than off-diagonal entries, its value is minimized over $\mathcal{Q}(p_0, \delta)$ by boundary points with $\|q - p_0\|_1 = \delta$.

Proof. By Corollary E.1.5, it suffices to restrict the middle infimum to boundary targets

$$q_{a,b} = p_0 + \frac{\delta}{2}(\mathbf{e}_a - \mathbf{e}_b), \qquad a \neq b.$$

For each such target, Lemma E.1.2 shows[1] that the inner supremum admits an optimizer of path-flow form

$$\frac{\delta}{2}\sum_{i=0}^{\kappa-2}\left(\mathbf{e}_{\sigma^i(a)}\mathbf{e}_{\sigma^{i+1}(a)}^\top - \mathbf{e}_{\sigma^{i+1}(a)}\mathbf{e}_{\sigma^{i+1}(a)}^\top\right) + \mathrm{diag}(p_0),$$

where $\sigma \in S_n$, $\kappa = \inf\{k \in \mathbb{Z}_+ : \sigma^k(a) = a\}$, and the path endpoint satisfies $\sigma^{\kappa-1}(a) = b$. Define

$$\begin{aligned} J(r,w,q) &= \sum_{v,s\in\mathcal{V}} w(v,s)\cdot\log r(v,s) \\ J^* &= \sup_r \inf_{q\in\mathcal{Q}_{\mathrm{ext}}(p_0,\delta)} \sup_{w\in\mathcal{P}_{\mathrm{flow}}(p_0,q)} J(r,w,q) \\ R^* &= \{r : \inf_{q\in\mathcal{Q}_{\mathrm{ext}}(p_0,\delta)} \sup_{w\in\mathcal{P}_{\mathrm{flow}}(p_0,q)} J(r,w,q) = J^*\} \\ r^* &= \arg\sup\{\mathrm{tr}[r] : r \in R^*\}. \end{aligned}$$

We say a solution $(\bar{w},\bar{q})$ is active if

$$\begin{aligned} \bar{w} &= \arg\sup_{w\in\mathcal{P}_{\mathrm{flow}}(p_0,\bar{q})} J(r^*,w,\bar{q}) \\ \bar{q} &= \arg\inf_{q\in\mathcal{Q}_{\mathrm{ext}}(p_0,\delta)} \sup_{w\in\mathcal{P}_{\mathrm{flow}}(p_0,q)} J(r^*,w,q) \\ J(\bar{w},\bar{q}) &= J^*. \end{aligned}$$

Fix $v \neq s \in \mathcal{V}$ and consider the perturbation for sufficiently small $\epsilon > 0$

$$\begin{aligned} \bar{r}(v,v) &\leftarrow r^*(v,v) + \epsilon \\ \bar{r}(v,s) &\leftarrow r^*(v,s) - \epsilon \\ \bar{r}(v',s') &\leftarrow r^*(v',s'),\ \forall (v',s') \neq (v,s). \end{aligned}$$

Then $\bar{r}$ cannot be a valid solution since it has higher trace than r^*. It follows that

$$\exists \text{ an active pair } (\bar{w},\bar{q}) \text{ such that} \quad \frac{dJ(\bar{r},\bar{w},\bar{q})}{d\epsilon} = \frac{\bar{w}(v,v)}{r^*(v,v)} - \frac{\bar{w}(v,s)}{r^*(v,s)} \leq 0.$$

Notice that the derivative $\frac{dJ(\bar{r},\bar{w},\bar{q})}{d\epsilon}$ must be either

[1]Since the seed alphabet has no canonical identification with the token alphabet, we work modulo a common relabelling of the random seeds. Then, after relabelling the seed alphabet by the same permutation, row-stochasticity, feasibility of the coupling, and the log-growth objective are unchanged. Thus we may assume WLOG that candidate r satisfies the diagonal dominance condition of Lemma E.1.1.

- No-transport: $\bar{w}(v,v) = p_0(v), \bar{w}(v,s) = 0$ so $\frac{dJ(\bar{r},\bar{w},\bar{q})}{d\epsilon} = \frac{p_0(v)}{r^*(v,v)} > 0$.
- Middle-way: $\bar{w}(v,v) = p_0(v) - \delta/2, \bar{w}(v,s) = \delta/2$ so $\frac{dJ(\bar{r},\bar{w},\bar{q})}{d\epsilon} = \frac{p_0(v)-\delta/2}{r^*(v,v)} - \frac{\delta/2}{r^*(v,s)}$.
- Transport-start: $\bar{w}(v,v) = p_0(v), \bar{w}(v,s) = \delta/2$ so $\frac{dJ(\bar{r},\bar{w},\bar{q})}{d\epsilon} = \frac{p_0(v)}{r^*(v,v)} - \frac{\delta/2}{r^*(v,s)}$.
- Transport-end: $\bar{w}(v,v) = p_0(v) - \delta/2, \bar{w}(v,s) = 0$ so $\frac{dJ(\bar{r},\bar{w},\bar{q})}{d\epsilon} = \frac{p_0(v)-\delta/2}{r^*(v,v)} > 0$.

Since $\frac{dJ(\bar{r},\bar{w},q)}{d\epsilon} \leq 0$, we can rule out the first and last cases. Now we can summarize that for any $v \neq s \in \mathcal{V}$ we have either Middle-way: $\bar{w}(v,v) = p_0(v) - \delta/2, \bar{w}(v,s) = \delta/2$ or Transport-start: $\bar{w}(v,v) = p_0(v), \bar{w}(v,s) = \delta/2$, and in any case $\frac{p_0(v)-\delta/2}{r^*(v,v)} - \frac{\delta/2}{r^*(v,s)} \leq 0$ holds.

Since $p_0(v) > \delta$, in either Middle-way or Transport-start case we have

$$0 \geq \frac{p_0(v) - \delta/2}{r^*(v,v)} - \frac{\delta/2}{r^*(v,s)} > \frac{\delta/2}{r^*(v,v)} - \frac{\delta/2}{r^*(v,s)}.$$

It follows that $r^*(v,v) > r^*(v,s)$. Since this argument holds for all $v \neq s \in \mathcal{V}$, r^* must satisfy $r^*(v,v) > r^*(v,s)$ for all $v \neq s \in \mathcal{V}$.

Next, we show that all active $\bar{q}, \bar{w}$ can be written as $\bar{q} = p_0 + \frac{\delta}{2} \cdot (\mathbf{e}_v - \mathbf{e}_s), \bar{w} = \frac{\delta}{2} \cdot (\mathbf{e}_v\mathbf{e}_s^\top - \mathbf{e}_s\mathbf{e}_s^\top) + \mathrm{diag}(p_0)$ for some $v \neq s \in \mathcal{V}$.

In either Middle-way or Transport-start case, there exists $a \neq b \in \mathcal{V}$ and $\sigma \in S_n$ and $i \in \mathbb{Z}$ such that $\bar{q} = p_0 + \frac{\delta}{2} \cdot (\mathbf{e}_a - \mathbf{e}_b)$ and $v = \sigma^i(a), s = \sigma^{i+1}(a)$, due to Lemma E.1.2. By Lemma E.1.4, with respect to $\bar{q} = p_0 + \frac{\delta}{2} \cdot (\mathbf{e}_v - \mathbf{e}_s)$, the optimizer of the inner problem must be written as

$$\frac{\delta}{2} \cdot (\mathbf{e}_v\mathbf{e}_s^\top - \mathbf{e}_s\mathbf{e}_s^\top) + \mathrm{diag}(p_0) = \arg \sup_{w \in \mathcal{P}_{\mathrm{flow}}(p_0,\bar{q})} J(r^*, w, \bar{q}).$$

Repeating this perturbation separately for each ordered pair (v,s) with $v \neq s$ yields, for each such pair, at least one active boundary coupling that witnesses the nonpositive directional derivative. The subsequent case analysis is applied to that witnessing active pair.

We can now explicitly write:

$$\begin{aligned}\inf_{q \in \mathcal{Q}_{\mathrm{ext}}(p_0,\delta)} \sup_{w \in \mathcal{P}_{\mathrm{flow}}(p_0,q)} J(r^*, w, q) &= \inf_{q \in \mathcal{Q}_{\mathrm{ext}}(p_0,\delta)} \sup_{w \in \mathcal{P}_{\mathrm{flow}}(p_0,q)} \sum_{v,s \in \mathcal{V}} w(v,s) \cdot \log r^*(v,s) \\ &= \inf_{v \neq s \in \mathcal{V}} \sum_{x \in \mathcal{V}} p_0(x) \log r^*(x,x) - \frac{\delta}{2} \cdot (\log r^*(s,s) - \log r^*(v,s))\end{aligned}$$

Define $J(r,v,s) = \sum_{x \in \mathcal{V}} p_0(x) \log r(x,x) - \frac{\delta}{2} \cdot (\log r(s,s) - \log r(v,s))$, we claim that $J(r^*,v,s)$ must be the same for all $v \neq s \in \mathcal{V}$. Otherwise suppose

$$J(r^*, \bar{v}, \bar{s}) = \sup_{v,s} J(r^*, v, s) > J^*.$$

Set

$$\begin{aligned}\bar{r}(\bar{v},\bar{v}) &\leftarrow r^*(\bar{v},\bar{v}) + \epsilon \\ \bar{r}(\bar{v},\bar{s}) &\leftarrow r^*(\bar{v},\bar{s}) - \epsilon \\ \bar{r}(v',s') &\leftarrow r^*(v',s'),\ \forall (v',s') \neq (\bar{v},\bar{s})\end{aligned}$$

for sufficiently small $\epsilon > 0$. Then for any ordered pair (v,s) with $v \neq s$ and $(v,s) \neq (\bar{v},\bar{s})$, the off-diagonal entry $r(v,s)$ is unchanged, so

$$\frac{dJ(\bar{r},v,s)}{d\epsilon} \geq \frac{p_0(\bar{v}) - \delta/2}{r^*(\bar{v},\bar{v})} \geq 0,$$

and thus

$$\inf_{q\in\mathcal{Q}_{\text{ext}}(p_0,\delta)} \sup_{w\in\mathcal{P}_{\text{flow}}(p_0,q)} J(r^*,w,q) = \inf_{v\neq s\in\mathcal{V},(v,s)\neq(\bar{v},\bar{s})} J(r^*,v,s) \geq J^*.$$

But $\text{tr}[\bar{r}] > \text{tr}[r^*]$, this is a contradiction.

From the last argument, we know that the optimal solution r^* is of the form

$$r^*_{a,b}(v,s) = \begin{cases} a, & s = v, \\ b, & s \neq v, \end{cases} \qquad a \geq 0,\ b \geq 0, \quad a + (n-1)b = 1,$$

with $a > b$ (strict diagonal dominance). We now optimize over the two parameters (a,b) subject to this constraint:

$$\sup_{a,b}\ \Phi(a,b) \quad \text{s.t. } a + (n-1)b = 1,\ a > 0,\ b > 0,\ a > b.$$

straightforward algebra shows the optimal objective value:

$$\begin{aligned}\Phi(a^*,b^*) &= (1-\tfrac{\delta}{2})\log a^* + \tfrac{\delta}{2}\log b^* \\ &= (1-\tfrac{\delta}{2})\log\ \ 1-\frac{\delta}{2}\Big) + \tfrac{\delta}{2}\log\ \ \frac{\delta}{2(n-1)}\Big).\end{aligned}$$

attained at $a^* = 1 - \frac{\delta}{2}$ and $b^* = \frac{\delta}{2(n-1)}$. Thus, the optimal value is

$$J^* = (1-\tfrac{\delta}{2})\log\ \ 1-\frac{\delta}{2}\Big) + \tfrac{\delta}{2}\log\ \ \frac{\delta}{2(n-1)}\Big).$$

This completes the proof.

□

Lemma E.1.7 (Row Normalization)**.** *Let $\mathcal{V} = \{1, \ldots, n\}$ and $p_0 \in \Delta(\mathcal{V})$ be a distribution over $\mathcal{V}$ such that $\inf_{v\in\mathcal{V}} p_0(v) > \delta$. Consider the problem*

$$\begin{aligned}\sup_{e} \inf_{q\in\mathcal{Q}(p_0,\delta)} \sup_{w\in\mathcal{P}(p_0,q)} & \sum_{v,s\in\mathcal{V}} w(v,s)\cdot \log e(v,s) \\ s.t. & \sum_{v,s\in\mathcal{V}} q(v)p_0(s)e(v,s) \le 1,\ \forall q\in\Delta(\mathcal{V}) \\ & e(v,s)\ge 0,\ \forall v,s\in\mathcal{V}\end{aligned}$$

where

$$\begin{aligned}\mathcal{Q}(p_0,\delta) =&\ \{q\in\Delta(\mathcal{V}) : \|q-p_0\|_1 \le \delta\} \\ \mathcal{P}(p_0,q) =&\ \left\{w(v,s) : \sum_s w(v,s) = q(v), \sum_v w(v,s) = p_0(s)\right\}\end{aligned}$$

Now define the kernel matrix $r : \mathcal{V}\times\mathcal{V}\mapsto\mathbb{R}$ such that each of its entries are defined as

$$r(v,s) = \frac{p_0(s)e(v,s)}{A(v)},\quad \forall v,s\in\mathcal{V},$$

where $A(v) := \sum_s p_0(s)e(v,s)$. Then $A^*(v) = \sum_s p_0(s)e^*(v,s) = 1$ at the optimizer e^*.

Proof. Let $e(v,s)$ be any feasible solution to the optimization problem. We define the scaling factor of e at node v as:

$$A(v) := \sum_{s\in\mathcal{V}} p_0(s)e(v,s).$$

Since we are maximizing an objective involving $\log e(v,s)$, we can assume $e(v,s) > 0$ strictly (otherwise the objective is $-\infty$), and consequently $A(v) > 0$.

We can decompose the matrix $e(v,s)$ into a scale-independent "shape" matrix $\bar{e}(v,s)$ and the scaling factors $A(v)$ as follows:

$$e(v,s) = A(v)\cdot\bar{e}(v,s),\quad \text{where } \bar{e}(v,s) = \frac{e(v,s)}{A(v)}.$$

By construction, the normalized matrix $\bar{e}$ satisfies the normalization property:

$$\sum_{s\in\mathcal{V}} p_0(s)\bar{e}(v,s) = \sum_{s\in\mathcal{V}} p_0(s)\frac{e(v,s)}{A(v)} = \frac{1}{A(v)}\sum_{s\in\mathcal{V}} p_0(s)e(v,s) = 1.$$

Now, let us analyze the constraint given in the problem statement. The condition is:

$$\sum_{v,s\in\mathcal{V}} q(v)p_0(s)e(v,s)\le 1,\quad \forall q\in\Delta(\mathcal{V}).$$

Substituting the decomposition of $e(v, s)$:

$$\sum_{v\in\mathcal{V}} q(v) \quad \sum_{s\in\mathcal{V}} p_0(s)e(v,s)\Bigg) = \sum_{v\in\mathcal{V}} q(v)A(v) \leq 1, \quad \forall q \in \Delta(\mathcal{V}).$$

Therefore $A(v) \leq 1$ for any $v \in \mathcal{V}$.

Next, we substitute the decomposition into the objective function. Using the property that $w \in \mathcal{P}(p_0, q)$ implies $\sum_s w(v, s) = q(v)$, we have:

$$\begin{aligned}\sum_{v,s\in\mathcal{V}} w(v,s)\log e(v,s) &= \sum_{v,s\in\mathcal{V}} w(v,s)\log\left(A(v)\cdot\bar{e}(v,s)\right)\\ &= \sum_{v,s\in\mathcal{V}} w(v,s)\log\bar{e}(v,s) + \sum_{v,s\in\mathcal{V}} w(v,s)\log A(v)\\ &= \sum_{v,s\in\mathcal{V}} w(v,s)\log\bar{e}(v,s) + \sum_{v\in\mathcal{V}} q(v)\log A(v).\end{aligned}$$

The first term depends only on the normalized shape $\bar{e}$, while the second term depends only on the scaling factors $A(v)$. To maximize the total objective, we must maximize the second term subject to the feasibility constraint derived above.

Consider the term $\sum_{v\in\mathcal{V}} q(v)\log A(v)$. Since the logarithm is a concave function, we can apply Jensen's inequality:

$$\sum_{v\in\mathcal{V}} q(v)\log A(v) \leq \log \quad \sum_{v\in\mathcal{V}} q(v)A(v)\Bigg).$$

From the feasibility constraint, we know that $\sum_{v\in\mathcal{V}} q(v)A(v) \leq 1$. Therefore:

$$\sum_{v\in\mathcal{V}} q(v)\log A(v) \leq \log(1) = 0.$$

Thus for every q,

$$\sup_{w\in\mathcal{P}(p_0,q)} \sum_{v,s} w(v,s)\log e(v,s) \leq \sup_{w\in\mathcal{P}(p_0,q)} \sum_{v,s} w(v,s)\log\bar{e}(v,s).$$

And the inequality is strict unless $\sum_v q(v)A(v) = 1$ and $A(v)$ is constant on the support of q.

Taking the minimum over q, we conclude:

$$\inf_q \sup_w \sum_{v,s} w(v,s)\log e(v,s) \leq \inf_q \sup_w \sum_{v,s} w(v,s)\log\bar{e}(v,s).$$

Equality is achieved if and only if $A(v) = 1$ for all $v \in \mathcal{V}$. Thus, for any optimal solution e^*, the scaling factors must be set to 1. Consequently:

$$A^*(v) = \sum_{s\in\mathcal{V}} p_0(s)e^*(v,s) = 1.$$

□

E.2 Proof of Theorem 6.4.3

Proof. We first prove the converse. Fix any valid e-value e. If e has zero or infinite entries that make the stopping time infinite under the adversary below, the lower bound is immediate; otherwise the log-increments are bounded. By the definition of J^* as the supremum over e-values, for every $\varepsilon > 0$ there exists $q_\varepsilon \in \mathcal{Q}(p_0, \delta)$ such that

$$\sup_{w \in \mathcal{P}(p_0, q_\varepsilon)} \sum_{v,s} w(v,s) \log e(v,s) \le J^* + \varepsilon.$$

Let the adversary choose $q^t \equiv q_\varepsilon$ at every time. For any generator, with $Y_t = \log e(v^t, s^t)$ and $S_t = \sum_{i=1}^t Y_i$,

$$\mathbb{E}[Y_t \mid \mathcal{F}_{t-1}] \le J^* + \varepsilon \qquad \text{a.s.}$$

Applying Theorem E.2.1 with $B_\alpha = \log(1/\alpha)$ gives

$$\inf_{\mathcal{G}} \liminf_{\alpha \downarrow 0} \frac{\mathbb{E}_{\mu(\mathcal{A}_\varepsilon, \mathcal{G})}[\tau_\alpha(e)]}{\log(1/\alpha)} \ge \frac{1}{J^* + \varepsilon}.$$

Taking the supremum over adversaries and then letting $\varepsilon \downarrow 0$ proves the lower bound.

For achievability, let the detector use e^* from Theorem 6.4.1. For every conditional target $q^t \in \mathcal{Q}(p_0, \delta)$, the generator chooses a maximal coupling of q^t and p_0. By Theorem 6.4.1,

$$\mathbb{E}[\log e^*(v^t, s^t) \mid \mathcal{F}_{t-1}] \ge J^* \qquad \text{a.s. for all } t.$$

Applying Theorem E.2.2 yields, for every adversary $\mathcal{A}$,

$$\inf_{\mathcal{G}} \limsup_{\alpha \downarrow 0} \frac{\mathbb{E}_{\mu(\mathcal{A}, \mathcal{G})}[\tau_\alpha(e^*)]}{\log(1/\alpha)} \le \frac{1}{J^*}.$$

Taking the supremum over $\mathcal{A}$ proves the achievability claim. □

E.2.1 Useful results

Theorem E.2.1 (Dynamic hitting-time lower bound with bounded upper drift)**.** *Let $(Y_t)_{t \ge 1}$ be adapted integrable random variables with $|Y_t| \le M$ almost surely, and let $S_t = \sum_{i=1}^t Y_i$. Suppose that, for some $J > 0$,*

$$\mathbb{E}[Y_t \mid \mathcal{F}_{t-1}] \le J \qquad \textit{a.s. for all } t.$$

For $B > 0$, define $\tau_B = \inf\{t \ge 1 : S_t \ge B\}$. Then

$$\mathbb{E}[\tau_B] = \infty \quad \textit{or} \quad \mathbb{E}[\tau_B] \ge \frac{B}{J}.$$

Consequently,

$$\liminf_{B \to \infty} \frac{\mathbb{E}[\tau_B]}{B} \ge \frac{1}{J}.$$

Proof. If $\mathbb{E}[\tau_B] = \infty$, there is nothing to prove. Otherwise $\tau_B < \infty$ almost surely and, since $|Y_t| \le M$,

$$\sum_{t\ge 1} \mathbb{E}[|Y_t|\mathbf{1}\{\tau_B \ge t\}] \le M\mathbb{E}[\tau_B] < \infty.$$

Thus Fubini's theorem and the drift assumption give

$$\mathbb{E}[S_{\tau_B}] = \sum_{t\ge 1} \mathbb{E}[Y_t\mathbf{1}\{\tau_B \ge t\}] \le J \sum_{t\ge 1} \mathbb{P}(\tau_B \ge t) = J\mathbb{E}[\tau_B].$$

Since $S_{\tau_B} \ge B$ by definition of τ_B, we obtain $B \le J\mathbb{E}[\tau_B]$. □

Theorem E.2.2 (Dynamic hitting-time upper bound with converging lower drift)**.** *Fix a filtered probability space $(\Omega, \mathcal{F}, (\mathcal{F}_t)_{t\ge 0}, \mathbb{P})$. Let $(Y_t)_{t\ge 1}$ be a sequence of integrable random variables adapted to $(\mathcal{F}_t)$, and define the partial sums*

$$S_t := \sum_{i=1}^{t} Y_i, \qquad t \ge 1,$$

with the convention $S_0 := 0$.

Assume the following.

*(A1) (*Bounded increments*) There exists a constant $M \in (0, \infty)$ such that*

$$|Y_t| \le M \quad \textit{almost surely for all } t \ge 1.$$

*(A2) (*Positive, converging lower conditional drift*) There exists a deterministic sequence $(J_t)_{t\ge 1}$ and constants $0 < J_{\inf} \le J_{\sup} < \infty$ such that*

$$\mathbb{E}[Y_t \mid \mathcal{F}_{t-1}] \ge J_t \quad \textit{almost surely for all } t \ge 1,$$

and

$$J_{\inf} \le J_t \le J_{\sup} \quad \textit{for all } t \ge 1,$$

with

$$J_t \longrightarrow J_\infty \in (0, \infty) \quad \textit{as } t \to \infty.$$

*(A3) (*Stopping rule*) For each threshold $B > 0$, define the stopping time*

$$\tau_B := \inf\{t \ge 1 : S_t \ge B\},$$

with the convention $\inf \emptyset := +\infty$.

Then for every $B > 0$, the stopping time τ_B is integrable. Moreover, for every $\varepsilon \in (0, J_\infty)$ there exists a finite constant C_ε such that

$$\mathbb{E}[\tau_B] \leq \frac{B + M + C_\varepsilon}{J_\infty - \varepsilon} \quad \textit{for all } B > 0.$$

Consequently,

$$\limsup_{B\to\infty} \frac{\mathbb{E}[\tau_B]}{B} \leq \frac{1}{J_\infty}.$$

Equivalently, if $B_\alpha := \log(1/\alpha)$ and

$$\tau_\alpha := \inf\{t \geq 1 : S_t \geq B_\alpha\}, \qquad \alpha \in (0, 1),$$

then

$$\limsup_{\alpha\downarrow 0} \frac{\mathbb{E}[\tau_\alpha]}{\log(1/\alpha)} \leq \frac{1}{J_\infty}.$$

Proof. Define the deterministic partial sums

$$A_t := \sum_{i=1}^{t} J_i, \qquad A_0 := 0,$$

and the excess process

$$Z_t := S_t - A_t = \sum_{i=1}^{t} (Y_i - J_i), \qquad Z_0 := 0.$$

By assumption (A2),

$$\mathbb{E}[Z_t \mid \mathcal{F}_{t-1}] = Z_{t-1} + \mathbb{E}[Y_t - J_t \mid \mathcal{F}_{t-1}] \geq Z_{t-1} \quad \text{a.s.},$$

so (Z_t) is a submartingale.

For $n \in \mathbb{N}$ define the bounded stopping time

$$\tau_B^{(n)} := \tau_B \wedge n.$$

We claim that for every n,

$$S_{\tau_B^{(n)}} \leq B + M \quad \text{a.s.}$$

Indeed, on $\{\tau_B \leq n\}$ we have $\tau_B^{(n)} = \tau_B$ and $S_{\tau_B - 1} < B$ by definition of τ_B, while $Y_{\tau_B} \leq M$ a.s., hence

$$S_{\tau_B} = S_{\tau_B - 1} + Y_{\tau_B} < B + M.$$

On $\{\tau_B > n\}$ we have $\tau_B^{(n)} = n$ and $S_n < B$, so again $S_{\tau_B^{(n)}} < B + M$. This proves the claim.

Next, we show that

$$\mathbb{E}[Z_{\tau_B^{(n)}}] \geq 0,$$

or equivalently, $\mathbb{E}[S_{\tau_B^{(n)}}] \geq \mathbb{E}[A_{\tau_B^{(n)}}]$.

Since $\tau_B^{(n)} \leq n$,

$$Z_{\tau_B^{(n)}} = \sum_{t=1}^{\tau_B^{(n)}} (Y_t - J_t) = \sum_{t=1}^{n} \mathbf{1}\{\tau_B^{(n)} \geq t\}(Y_t - J_t).$$

For $t \leq n$, $\{\tau_B^{(n)} \geq t\} = \{\tau_B \geq t\} \in \mathcal{F}_{t-1}$, since τ_B is a stopping time. Therefore,

$$\begin{aligned}
\mathbb{E}\Big[\mathbf{1}\{\tau_B^{(n)} \geq t\}(Y_t - J_t)\Big] &= \mathbb{E}\Big[\mathbb{E}\Big[\mathbf{1}\{\tau_B^{(n)} \geq t\}(Y_t - J_t) \mid \mathcal{F}_{t-1}\Big]\Big] \\
&= \mathbb{E}\Big[\mathbf{1}\{\tau_B^{(n)} \geq t\}\mathbb{E}[Y_t - J_t \mid \mathcal{F}_{t-1}]\Big] \\
&\geq 0,
\end{aligned}$$

where the inequality follows from assumption (A2). Summing over $t = 1, \ldots, n$ yields $\mathbb{E}[Z_{\tau_B^{(n)}}] \geq 0$.

Thus, we have that

$$\mathbb{E}[A_{\tau_B^{(n)}}] \leq \mathbb{E}[S_{\tau_B^{(n)}}] \leq B + M \quad \text{for all } n.$$

Since $J_t \geq J_{\inf} > 0$, the sequence (A_t) is increasing and $A_t \to \infty$ as $t \to \infty$. Because $\tau_B^{(n)} \uparrow \tau_B$, the monotone convergence theorem yields

$$\mathbb{E}[A_{\tau_B}] = \lim_{n\to\infty} \mathbb{E}[A_{\tau_B^{(n)}}] \leq B + M.$$

Moreover, since $A_{\tau_B} \geq J_{\inf}\tau_B$,

$$J_{\inf}\mathbb{E}[\tau_B] \leq \mathbb{E}[A_{\tau_B}] \leq B + M,$$

so τ_B is integrable.

Fix $\varepsilon \in (0, J_\infty)$. Since $J_t \to J_\infty$, there exists $N = N(\varepsilon)$ such that

$$J_t \geq J_\infty - \varepsilon \quad \text{for all } t \geq N.$$

Define the finite constant

$$C_\varepsilon := \sup_{0 \leq t \leq N-1} \Big((J_\infty - \varepsilon)t - A_t\Big).$$

Then for all $t \geq 0$,

$$A_t \geq (J_\infty - \varepsilon)t - C_\varepsilon.$$

Applying the bound from in the previous display at the random time τ_B and taking expectations yields

$$\mathbb{E}[A_{\tau_B}] \geq (J_\infty - \varepsilon)\mathbb{E}[\tau_B] - C_\varepsilon.$$

Combining this with $\mathbb{E}[A_{\tau_B}] \leq B + M$ yields

$$(J_\infty - \varepsilon)\mathbb{E}[\tau_B] \leq B + M + C_\varepsilon,$$

and therefore

$$\mathbb{E}[\tau_B] \leq \frac{B + M + C_\varepsilon}{J_\infty - \varepsilon}.$$

Dividing by B and letting $B \to \infty$, then letting $\varepsilon \downarrow 0$, gives

$$\limsup_{B\to\infty} \frac{\mathbb{E}[\tau_B]}{B} \leq \frac{1}{J_\infty}.$$

This completes the proof. □

E.3 Useful Claims

Lemma E.3.1. *Let $\mathcal{V}$ be the set $\{1, \ldots, n\}$ and $p_0 \in \Delta(\mathcal{V})$ be a distribution over $\mathcal{V}$ such that $\min_{v\in\mathcal{V}} p_0(v) > \delta$. Define*

$$\mathcal{Q}(p_0, \delta) = \{q \in \Delta(\mathcal{V}) : \|q - p_0\|_1 \leq \delta\}$$

Then $\mathcal{Q}(p_0, \delta)$ is a convex polytope whose vertex set is given by:

$$\mathcal{Q}_{\text{ext}}(p_0, \delta) := \{p_0 + (\mathbf{e}_i - \mathbf{e}_j) \cdot \delta/2 : (i, j) \in \mathcal{V} \times \mathcal{V}, i \neq j\}.$$

Proof. Recall $\mathcal{Q}(p_0, \delta) := \{q \in \Delta^n : \|q - p_0\|_{\ell_1} \leq \delta\}$. First, we note that $\mathcal{Q}(p_0, \delta)$ is the intersection of two convex polytopes, hence it must also be a convex polytope. We claim that $\mathcal{Q}(p_0, \delta) = \text{Conv}(\mathcal{Q}_{\text{ext}}(p_0, \delta))$. Suppose $q \in \mathcal{Q}(p_0, \delta)$, then we can write that $q_i - p_i = s_i$ for $s_i \in [-\delta/2, \delta/2]$, $\sum s_i = 0$, and $\sum|s_i| \leq \delta$. Next, suppose for contradiction that $s_j > \delta/2$, then $\sum_{i\neq j} s_i = -s_j$ and hence,

$$\sum_k |s_k| \geq |s_j| + |\sum_{i\neq j} s_i| > \delta,$$

which violates the TV constraint.

Now we show that every $q \in \mathcal{Q}(p_0, \delta)$ lies in the convex hull of $\mathcal{Q}_{\text{ext}}(p_0, \delta)$. Write $s = q - p_0$ and set

$$\eta := \frac{\|q - p_0\|_1}{\delta} \in [0, 1].$$

Let $P = \{i : s_i > 0\}$ and $N = \{j : s_j < 0\}$. Choose nonnegative α_{ij} for $i \in P$, $j \in N$ such that

$$\sum_{j \in N} \alpha_{ij} = \frac{s_i}{\delta/2}, \qquad \sum_{i \in P} \alpha_{ij} = \frac{-s_j}{\delta/2}.$$

Then $\sum_{i,j} \alpha_{ij} = \eta$ and

$$p_0 + \frac{\delta}{2} \sum_{i \in P, j \in N} \alpha_{ij}(\mathbf{e}_i - \mathbf{e}_j) = q.$$

To make a convex combination, choose any symmetric probability mass ρ_{ij} on ordered pairs $i \neq j$, i.e., $\rho_{ij} = \rho_{ji}$ and $\sum_{i \neq j} \rho_{ij} = 1$. Then

$$\sum_{i \neq j} \rho_{ij} \quad p_0 + \frac{\delta}{2}(\mathbf{e}_i - \mathbf{e}_j)\Big) = p_0.$$

Define

$$\lambda_{ij} = \begin{cases} \alpha_{ij} + (1 - \eta)\rho_{ij}, & i \in P,\ j \in N, \\ (1 - \eta)\rho_{ij}, & \text{otherwise.} \end{cases}$$

The coefficients λ_{ij} are nonnegative, sum to one, and satisfy

$$q = \sum_{i \neq j} \lambda_{ij} \quad p_0 + \frac{\delta}{2}(\mathbf{e}_i - \mathbf{e}_j)\Big).$$

It is now enough to show that any $v \in \mathcal{Q}_{\text{ext}}(p_0, \delta)$ cannot be generated by a convex combination of two distinct points in $\mathcal{Q}(p_0, \delta)$ which will prove that $v \in \mathcal{Q}_{\text{ext}}(p_0, \delta)$ is a vertex of $\mathcal{Q}(p_0, \delta)$ and hence, $\mathcal{Q}(p_0, \delta)$ is generated by the convex hull of $\mathcal{Q}_{\text{ext}}(p_0, \delta)$. Suppose for contradiction that this is the case. Then there exists $q, q' \in \mathcal{Q}(p_0, \delta)$ and $\lambda \in (0, 1)$ such that

$$p_0 + (\mathbf{e}_i - \mathbf{e}_j)\frac{\delta}{2} = \lambda q + (1 - \lambda)q'.$$

Let $q = p_0 + \eta\frac{\delta}{2}$ and $q' = p_0 + \eta'\frac{\delta}{2}$ where $\eta \neq \eta'$ then we have that

$$\mathbf{e}_i - \mathbf{e}_j = \lambda\eta + (1 - \lambda)\eta'.$$

Then we must have that

$$\lambda\eta_i + (1 - \lambda)\eta'_i = 1, \quad \lambda\eta_j + (1 - \lambda)\eta'_j = -1,$$

and

$$\lambda\eta_k + (1 - \lambda)\eta'_k = 0.$$

But for either of the first two conditions to be true, we must have that $\eta_i = \eta'_i = 1$ and $\eta_j = \eta'_j = -1$ Since all elements η_k and η'_k are bounded in magnitude by 1. Furthermore, because $|\eta_i| + |\eta_j| = 2$, all other $\eta_k = 0$ (same for η'). Thus, $\eta = \eta'$ which is the desired contradiction. □

E.4 Additional Experiments

E.4.1 Full results of Section 6.5.1

This section summarizes the complete experiment results on the MarkMyWords benchmark [152] under three temperature settings $(0.3, 0.7, 1.0)$. For the generation scheme, we follow the hyperparameter configurations in Chapter 5 with $K = 20, |\Omega_h| = 2, \delta = 0.3$. Table E.1 reports generation quality across watermarking schemes. Overall, quality is relatively stable across schemes, and our method achieves quality that is comparable to (and in some cases close to the best among) the baselines at each temperature. Table E.2 reports detection size, measuring the median number of tokens needed to detect the watermark under a range of perturbations, where lower is better. Here, our scheme consistently attains the smallest size at every temperature, substantially outperforming prior methods. This indicates that our detector can reliably identify watermarked text using fewer tokens. Together, these results confirm that our e-value-based scheme improves detection efficiency without sacrificing generation quality.

Temp.	Exponential	Inverse Transform	Binary	Distribution Shift	SEAL	Our e-value scheme
0.3	0.906	**0.910**	0.905	0.900	0.905	0.909
0.7	0.907	0.917	**0.919**	0.912	0.901	**0.919**
1.0	0.898	**0.917**	0.905	0.907	0.871	0.902

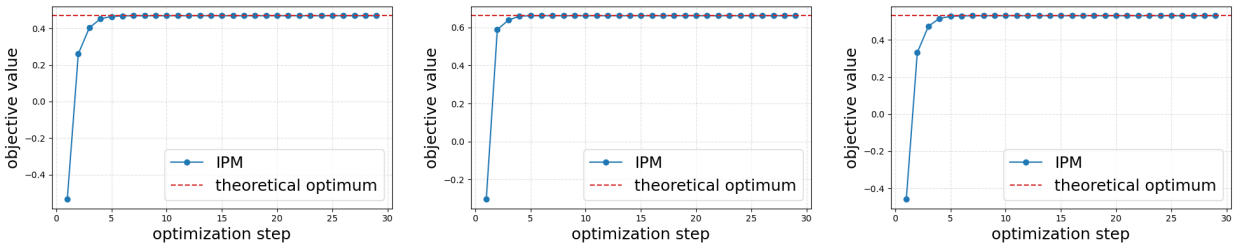
Table E.1: Comparison of our e-value scheme with baselines on quality (↑) across different temperature settings. Best (per temperature) is highlighted in bold.

Temp.	Exponential	Inverse Transform	Binary	Distribution Shift	SEAL	Our e-value scheme
0.3	∞	∞	∞	112.5	106.0	**97.0**
0.7	∞	734.0	∞	145.0	84.5	**72.0**
1.0	240.5	163.5	∞	317.0	133.0	**97.5**

Table E.2: Comparison of our e-value scheme with baselines on size (↓) across different temperature settings. Best (per temperature) is highlighted in bold.

E.4.2 Synthetic experiments

In this section, we verify our theoretical results with synthetic experiments on a family of simple target distributions. Here, we consider the two-token case where $|\mathcal{V}| = 2$ and for simplicity of notation, we let $\mathcal{V} = \{0, 1\}$. Any anchor distribution we consider belongs to a one-parameter Bernoulli family $p_0 = \begin{bmatrix} p & 1-p \end{bmatrix}$ for $p \in (0, 1)$.



(a) $p = 0.2$ (b) $p = 0.5$ (c) $p = 0.75$

Figure E.1: Simulation of the log growth rate in Eq. (6.2) in two-token case. Three separate anchor distributions are used each with parameter $\delta = 0.01$. We solve the simplified maxmin problem using the CLARABEL interior point method solver which is run for 30 steps. The theoretical optimum is computed as in Eq. (6.5).

Log-growth optimality. We solve the optimization problem in Eq. (6.2) numerically and compare the objective curve with the theoretical prediction. With $n = 2$, the problem can be simplified to the following maxmin problem:

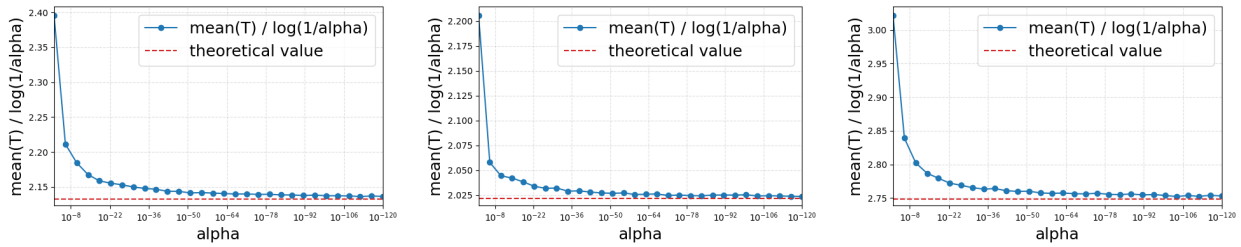
$$\sup_{e} \sum_{v} p_0(v) \log e(v, v) + \frac{\delta}{2} \min \left\{ \log \frac{e(0,1)}{e(1,1)}, \log \frac{e(1,0)}{e(0,0)} \right\}$$

subject to the constraint in Eq. (6.1) with $\mathcal{Q}(p_0, \delta)$ replaced with the vertex set

$$\mathcal{Q}_{\text{ext}}(p_0, \delta) := \left\{ p_0 \pm \frac{\delta}{2}(\mathbf{e}_0 - \mathbf{e}_1) \right\}.$$

Using the CLARABEL interior point method solver [67], we obtain the results in Fig. E.1. We choose $\delta = 0.01$ and $p = 0.2, 0.5, 0.75$ corresponding to three separate anchor distributions p_0. For each p_0, we cold start the IPM solver and run it for 30 steps with maximum allowed step size 0.99. We observe that our numerical solution to Eq. (6.2) converges to the proposed theoretical value in Eq. (6.5), hence verifying our theoretical results in the two-token case.

Stopping time. In this setting, we simulate sequential testing with the optimal e-value e^*, as in Eq. (6.3). Following Theorem 6.4.1, we let $\mathcal{A}^*$ be the adversary that chooses $q_t = q^*$ for all t and $\mathcal{G}^*$ be the generator that selects $w_t = w^*$ in response to $\mathcal{A}^*$ for all t. Under this setting, we simulate the stopping time over several α values to obtain the results in Fig. E.2. Here, we choose the parameters $\delta = 0.1$ and $p = 0.2, 0.5, 0.75$. For each anchor p_0, we compute the corresponding q^*, w^*, and e^* using the closed-form equations in Theorem 6.4.1. We select 30 values of α evenly log-spaced from 10^{-2} to 10^{-120}. For each α, we compute 10000 stopping-times by generating sequences $(V^t, S^t)_{t \geq 1} \sim w^*$ and computing the resulting $\log e^*(V^t, S^t)$. We then average the stopping times to form an estimate of $\mathbb{E}_{\mu(\mathcal{A}^*, \mathcal{G}^*)}[\tau_\alpha(e^*)]$. In Fig. E.2, we plot our estimates of $\mathbb{E}_{\mu(\mathcal{A}^*, \mathcal{G}^*)[\tau_\alpha(e^*)]}/(\log(1/\alpha))$ which show that as $\alpha \downarrow 0$, the estimates converge to the theoretical rate of $1/J^*$, thereby validating the result in Theorem 6.4.3.



(a) $p = 0.2$ (b) $p = 0.5$ (c) $p = 0.75$

Figure E.2: Simulation of the stopping time $\mathrm{SC}(e^*)$ in two-token case. We simulate for three different anchor distributions $p_0 = (p, 1-p)$ with $\delta = 0.1$ and estimate average stopping times $\mathbb{E}[\tau_\alpha]$ for α values ranging from 10^{-2} to 10^{-120}. Each $\mathbb{E}[\tau_\alpha]$ is estimated by simulating 10000 stopping times τ_α. The plots above display graphs of $\frac{\mathbb{E}[\tau_\alpha]}{\log(1/\alpha)}$ with red dashed lines equal to $1/J^*$ where J^* is as in Eq. (6.5). We observe that convergence to the theoretical optimum is obtained for sufficiently small α.

E.4.3 Empirical validity of the anchor distribution assumption

Anchored E-Watermarking assumes that the target distribution q lies in the ℓ_1 ball $\mathcal{Q}(p_0, \delta)$ around the anchor and that p_0 satisfies $\inf_{v \in \mathcal{V}} p_0(v) \geq \delta$. To examine whether these conditions hold for realistic LLM pairs, we measure the relevant quantities on the alphabet at which our scheme is actually instantiated. Following the green–red list construction of Kirchenbauer et al. [99], the scheme operates not on the raw token vocabulary but on a partition of $\mathcal{V}$ into fewer than five groups, on which probability masses remain bounded away from zero. This justifies the use of a moderate $\delta > 0$ even when the minimum token-level probability is close to 0.

Table E.3 reports the median ℓ_1 distance between the partition-level target (Llama2-7B-chat) and anchor (Phi-3-mini-128k-instruct) distributions, together with the median of $\min_v p_0(v)$, evaluated on the MarkMyWords prompts.

Temp.	**Median ℓ_1 distance**	**Median $\min_v p_0(v)$**
0.3	0.32	0.33
0.7	0.24	0.30
1.0	0.21	0.33

Table E.3: Partition-level statistics of the target and anchor distributions on MarkMyWords.

The values are consistent with the intuition that higher-temperature decoding yields more diffuse distributions, simultaneously shrinking the ℓ_1 gap and lifting $\min_v p_0(v)$ away from zero. The radius δ is a robustness/power parameter: increasing it enlarges $\mathcal{Q}(p_0, \delta)$ but

decreases the worst-case log-growth rate J^* in Theorem 6.4.1. When the configured radius underestimates the empirical gap, Ville's inequality still guarantees Type-I error control, but the power and stopping-time guarantees from Theorem 6.4.1 and Theorem 6.4.3 may degrade.

E.4.4 Empirical Type-I error

Type-I error control is theoretically guaranteed by the validity of the e-value and Ville's inequality, but it is informative to verify the guarantee empirically. We evaluate Theorem 6.4.1 on MarkMyWords at the nominal level $\alpha = 0.02$ used in the main experiments, running the detector on unwatermarked text generated from the same prompts.

Temp.	Type-I error
0.3	0.00
0.7	0.00
1.0	0.00

Table E.4: Empirical Type-I error of our e-value scheme on MarkMyWords at $\alpha = 0.02$. The observed false-positive rate is zero at every temperature. For the optimal e-value the one-step null expectation is exactly one for every token value, so the empirical slack is due to the conservativeness of the finite-time Ville threshold, not to a restriction of the null distribution to $\mathcal{Q}(p_0, \delta)$.

The observed false-positive rate is strictly below $\alpha = 0.02$ in every setting. This slack does not come from looseness of the one-step null expectation: for the optimal e-value,

$$\sum_{s \in \mathcal{V}} p_0(s) e^*(v, s) = 1, \qquad \forall v \in \mathcal{V},$$

and therefore

$$\mathbb{E}_{v \sim q,\, s \sim p_0}[e^*(v, s)] = 1, \qquad \forall q \in \Delta(\mathcal{V}).$$

The gap between the observed false-positive rate and α instead reflects the conservativeness of Ville's inequality for finite-time threshold crossings and the finite inspection budget used in the experiments.

E.4.5 Additional target model

The main experiments in Section 6.5.1 use Llama2-7B-chat as the target model. We additionally instantiate the target with Qwen3-4B [212], while keeping the decoding and detection parameters fixed, to assess whether the sample-efficiency gains transfer across model families. Table E.5 reports the size of Theorem 6.4.1 against the same five baselines as in the main text. As in Table 6.1, an entry of ∞ indicates that more than half of the watermarked generations fail to be detected within the inspection budget.

Temp.	Exponential	Inverse Transform	Binary	Distribution Shift	SEAL	Our e-value scheme
0.3	∞	∞	∞	**111.0**	128.5	127.5
0.7	400.0	232.0	∞	151.0	110.0	**80.0**
1.0	207.5	169.0	∞	272.5	149.0	**129.5**

Table E.5: Comparison of our e-value scheme with baselines on size (↓), using Qwen3-4B as the target model and Qwen3-1.7B as the anchor. Best (per temperature) is highlighted in bold.

The pattern mirrors the Llama2-7B-chat results: Theorem 6.4.1 attains the smallest size at temperatures 0.7 and 1.0, and remains competitive with the best baseline at temperature 0.3. The efficiency benefits of the optimal anchored e-value therefore do not depend on a specific target model.

E.4.6 Comparison against chunk-based sequential testing baselines

We compare with another competitive sequential baseline which restricts the detector to a finite, pre-declared set of inspection times and applies a tighter Bonferroni correction over that set. Concretely, fix a chunk size L and a chunk count M, defining an inspection budget of ML tokens. For each baseline, the detector evaluates the cumulative p-value p_k only at the chunk endpoints $k \in \{L, 2L, \ldots, ML\}$ and applies the fixed correction $p_k \mapsto M \cdot p_k$. Detection is declared at the first endpoint $k = mL$ for which $M \cdot p_{mL} < \alpha$, equivalently $p_{mL} < \alpha/M$, and the reported size is therefore an integer multiple of L. By union bound applied to the M inspection times, this procedure controls Type-I error at level α. In our experiments we set $M = 20$ and $L = 20$, giving an inspection budget of 400 tokens.

Temp.	Exponential	Inverse Transform	Binary	Distribution Shift	SEAL	Our e-value scheme
0.3	∞	∞	∞	100.0	100.0	**97.0**
0.7	260.0	380.0	∞	100.0	80.0	**72.0**
1.0	130.0	200.0	∞	160.0	100.0	**97.5**

Table E.6: Comparison of our e-value scheme with baselines on size (↓) under the chunk-based sequential protocol with $M = 20$ chunks of size $L = 20$ tokens (correction $p \mapsto M \cdot p$) on MarkMyWords. Best (per temperature) is highlighted in bold.

The chunk-based correction tightens several baselines relative to the per-token Bonferroni results of Table E.2. Nevertheless, Theorem 6.4.1 remains the most efficient method at every temperature.