To Theoretically Understand Transformer-Based In-Context Learning for Optimizing CSMA

Shugang Hao†, Hongbo Li†, and Lingjie Duan
Singapore University of Technology and Design
Singapore, Singapore
shugang_hao@sutd.edu.sg,hongbo_li@mymail.sutd.edu.sg,lingjie_duan@sutd.edu.sg

Abstract

Even with enhanced throughput and low latency in WiFi 7, the heavy contention among many wireless devices in unlicensed bands can still trigger severe collisions under dynamic channel environments with unknown and varying node densities. Traditional model-based approaches (e.g., non-persistent and p-persistent CSMA) simply optimize the backoff strategies under a known and fixed contention node density. This paper is the first to introduce and analyze in-context learning (ICL) for optimizing CSMA. We first prove that the traditional model-based approaches lead to a large throughput loss due to an inaccurate estimation of the channel environment parameters. Then we propose our transformer-based ICL optimizer containing four steps: ICL data collection, prompt construction, embedding, and transformer training. We construct prompts as the input to the transformer by pre-collecting multiple points (of the current collision parameters and the corresponding contention window thresholds) and a query collision case. To train the transformer, we employ gradient descent to develop an efficient algorithm to analyze its learning convergence. Our analysis guarantees that within a limited number of steps, the transformer accurately predicts a near-optimal contention window threshold for any given query token, ensuring a converged throughput loss. We further extend to consider erroneous data input examples with erroneous contention window thresholds to the transformer. We prove that our transformer-based ICL optimizer still incurs limited ICL prediction and throughput losses to the optimum. Experimental results further demonstrate our approach's great advantage over benchmarks to well adapt to unknown and large node densities.

CCS Concepts

• **Networks → Link-layer protocols**; **Wireless local area networks**; • **Computing methodologies → Machine learning algorithms**.

† Both authors contributed equally to this work.

MobiHoc'25, Houston, TX

ACM ISBN 978-1-4503-XXXX-X/2018/06
https://doi.org/XXXXXXX.XXXXXXX

Keywords

Transformer, in-context learning, CSMA, dynamic node density, convergence analysis

ACM Reference Format:
Shugang Hao†, Hongbo Li†, and Lingjie Duan. 2018. To Theoretically Understand Transformer-Based In-Context Learning for Optimizing CSMA. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (MobiHoc'25).* ACM, New York, NY, USA, 21 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Even with enhanced throughput and low latency in WiFi 7, the heavy contention in unlicensed bands can still trigger severe collisions when many devices co-use the same channel in dynamic network environments (e.g., [11], [5], [7]). To reduce collisions, the binary exponential backoff (BEB) scheme widly in use (e.g., WiFi 6 [14] and WiFi 7 [8]) asks each device or node to wait for a random period whenever a collision occurs, where the contention window threshold is doubled after each collision. In dynamic channel environments, it is hard to estimate the node density in a distributed BEB for determining contention window thresholds, resulting in a degraded throughput.

In the CSMA literature, traditional model-based approaches (e.g., non-persistent and p-persistent CSMA in [6], [5], [1], [25], [10]) optimize the backoff strategies under a known node density assumption. Knowing the node density, they derive the throughput formulation in closed forms for maximization and solve the optimal contention window thresholds in terms of the node density. Their assumption no longer holds under dynamic channel environments with unknown or varying node densities. One question arises:

- *Q1. How well can a model-based approach adapt to dynamic channel environments with unknown node densities?*

Later we prove that any model-based approach can lead to a large throughput loss.

There are also deep reinforcement learning (DRL) based model-free approaches for unknown environments in the recent CSMA literature (e.g., [3], [29], [30]). In particular, Wydmański *et al.* (2021) in [29] propose a DRL approach to dynamically adjust contention window size based on turn-around-time measurement of channel status in IEEE 802.11ax networks. Yan *et al.* (2024) in [30] presents a multi-device distributed DRL framework that intelligently tunes the contention window in IEEE 802.11bn networks to minimize tail latency while maintaining throughput. Note that all these approaches still focus on fixed channel environment parameters like node density. Thus, models trained in one channel environment do not generalize well to other dynamic environments, still leading to a degraded throughput (e.g., [20], [16]).

A novel approach is to use a transformer to perform in-context learning (ICL). ICL refers to the incredible ability of a large language model (LLM) to understand and execute a task based solely on a few examples provided in the prompt, rather than through traditional training or fine-tuning (e.g., [9], [23], [21]). Unlike the costly DRL-based approaches, ICL does not change any network parameter once pre-trained and is widely adaptable to many complex tasks like mathematical reasoning problems (e.g., [18], [19]). Recently, the communication society has explored the direction of introducing ICL to network design, such as transmission power allocation (e.g., [15], [34]), network deployment (e.g., [2], [24]), and network detection (e.g., [26], [31]). All these studies are empirical, lacking a theoretical analysis with performance guarantees.

In the transformer-based ICL literature, there are a few studies on analyzing and providing theoretical guarantees of transformer-based ICL prediction (e.g., [32], [13], [17]). In particular, Zhang *et al.* (2024) in [32] investigate the training dynamics of linear transformers for ICL. Huang *et al.* (2024) in [13] give convergence guarantees of ICL based on a one-layer transformer with softmax attention and linear mapping functions. Li *et al.* (2024) in [17] further analyze the convergence bounds for binary classification problems. All these studies consider either binary output or a linear mapping function. In a practical CSMA design, the mapping function from the collision parameters to the contention window threshold is more complex and no longer linear. Further, contention window thresholds are defined on a set of integers instead of a simple binary set. Therefore, their approaches are not suitable for CSMA.

There are two challenges to design a transformer-based ICL optimizer for CSMA. First, ICL involves the non-linear transformer architecture with a softmax operation and its prediction is highly non-convex in the transformer parameters, making it challenging to optimize for a small training loss. Secondly, the channel environment changes from time to time, requiring our algorithm to be efficient with guaranteed convergence within a limited number of training steps. Consequently, our second question arises:

- *Q2. How to design an ICL-based approach for training the transformer parameters with a convergence guarantee?*

We summarize our key novelty and main results below.

- *Theoretical understanding of transformer-based in-context learning for optimizing CSMA:* To the best of our knowledge, we are the first to introduce and analyze ICL for optimizing CSMA. We focus on a typical non-persistent CSMA (NP-CSMA) adopting the distributed coordination function (DCF), which is widely used in 802.11 protocols in practice. Traditional model-based approaches (e.g., non-persistent and p-persistent CSMA) simply optimize the backoff strategies under a known and fixed contention node density, which we prove results in a large throughput loss. Meanwhile, existing DRL-based approaches in the recent CSMA literature still focus on fixed channel environment parameters like node density. Thus, models trained in one channel environment do not generalize well to other dynamic environments, still leading to a degraded throughput. To address these limitations, we provide new analytical insights into *how a system can leverage transformer-based ICL to optimize throughput.* Our approach eliminates the need for prior knowledge of node density and well adapts to unknown and varying ones.
- *Four-step transformer-based ICL optimizer design with bounded ICL training and throughput losses:* We propose our transformer-based ICL optimizer to contain four steps: ICL data collection, prompt construction, embedding, and transformer training. We construct prompts as the input to the transformer by pre-collecting multiple points (of the current collision parameters and the corresponding contention window thresholds) and a query collision case. To train the transformer, we employ gradient descent to develop an efficient algorithm to analyze its learning convergence. Subsequently, we analyze its convergence state and prove that, within a limited number of steps, the transformer accurately predicts the optimal contention window threshold for a given query collision number, ensuring a limited throughput loss.
- *Extension to the case of erroneous prompts:* In practice, it may be difficult for the system to obtain the optimal collision-threshold pairs to construct perfect prompts. We thus extend our system model to consider erroneous contention windows in the prompts. We still manage to prove that the ICL prediction loss is upper bounded by a constant, leading to a limited throughput loss. Experimental results further demonstrate our approach's great advantage over benchmarks to well adapt to unknown and large node densities.

The rest of this paper is organized as follows. Section 2 introduces the system model of a typical time-slotted NP-CSMA and the throughput optimization problem. Section 3 analyzes the model-based and DRL-based approaches in the CSMA literature as two benchmarks for our approach to compare later. Section 4 details our transformer-based ICL approach design for optimizing CSMA. Section 5 illustrates the analysis of our approach and gives theoretical guarantees of ICL convergence, the ICL training loss, and the throughput loss. Section 6 extends to consider the erroneous prompt case and prove the analytical bounds of the ICL prediction and the throughput losses. Section 7 conduct experiments to verify our theoretical results. Section 8 finally concludes the paper.

2 System Model and Problem Formulation

In this section, we first introduce our system model based on a typical NP-CSMA. Then, we formulate the throughput maximization problem for further analysis in Sections 3-6.

The distributed coordination function (DCF) for multiple contention nodes' channel access and collision avoidance has been widely deployed in practical 802.11 protocols (e.g., the latest WiFi-7 [8]). Based on DCF, we consider a general slotted NP-CSMA with N contention nodes. The principles of our NP-CSMA are as follows:

i) The system operates the NP-CSMA in a time-slotted way, where each node's transmission decision is only made at the beginning of each time slot $t \in [T] := \{1, \cdots, T\}$. It pre-determines the backoff strategy $\{(k, W_k)\}_{k=0}^{K}$ for all the nodes, where K denotes the maximum collision number to be allowed for each packet transmission, $k \in \{0, \cdots, K\}$ denotes the collision number since the last successful transmission and W_k is a positive integer to represent the contention window threshold for the collision number k. Following the

CSMA literature, $W_0 \leq \cdots \leq W_K$. Note that the above backoff strategy is more general than the BEB scheme widely used in 802.11 protocols, which is a special case with contention window threshold $W_k = 2^k W_0$ for $k \in [K]$.

ii) Each node $i \in [N]$ senses the co-used communication channel whenever it has a data packet to transmit during a time slot $t \in [T]$. If the channel is sensed as idle for a distributed interframe space (DIFS), it uses the initial contention window threshold W_0 to generate a random integer timer between 0 and $W_0 - 1$ to backoff before transmission. Otherwise, it does not transmit the packet and keeps sensing until the channel is idle for a DIFS.

iii) If its packet transmission incurs a collision, it draws a random integer timer between 0 and $W_k - 1$ according to the current collision number k to backoff. It then counts down the timer and transmits the packet until the timer reduces to 0.

The objective is to find the optimal contention window thresholds $\{W_k^*\}_{k=0}^K$ to maximize the throughput for successful packet transmissions. Following the CSMA literature (e.g., [1], [28], [5]), we define the throughput U as the fraction of time that the channel is used to successfully transmit a packet as follows:

$$U = \frac{\mathbb{E}[\text{A packet is successfully transmitted in a slot time}]}{\mathbb{E}[\text{Length of a slot time}]}. \quad (1)$$

The corresponding throughput maximization problem is given as

$$\max_{\{W_k\}_{k=0}^K} U(\{W_k\}_{k=0}^K) \text{ in (1)}.$$

To maximize the above throughput U, the traditional model-based approaches (e.g., [6], [5], [1], [25], [10]) rely on the exact formulation of U for solving the optimal contention window thresholds. They require the exact knowledge of the packet transmission probability P_{tr} and the successful transmission probability P_s, which are difficult to obtain in dynamic network environments.

For an improved design of ICL for CSMA, a converged loss of our approach's throughput $\hat{U}$ from that at the optimum U^* should be guaranteed within a limited number of training steps. We need to carefully design our transformer-based ICL approach to satisfy two desired properties as follows:

- *Optimality under dynamic node densities.* The proposed solution should approach the optimum contention window thresholds as close as possible under dynamic and unknown node densities after a limited number of training steps.
- *Robustness to erroneous data input.* The proposed solution should still guarantee a limited throughput loss even with erroneous data input.

In the rest of this work, for a vector w, we let $\|w\|$ denote its ℓ-2 norm. For some positive constant c_1 and c_2, we define $x = \Theta(y)$ if $c_1|y| < x < c_2|y|$ and $x = O(y)$ if $x < c_1|y|$. Let $\mathbb{N}$ denote the set of natural numbers.

3 Benchmarks: Model-Based and DRL-Based Approaches

In Section 3.1, we introduce a model-based approach as a benchmark and prove its inefficiency in adapting to unknown node densities. In Section 3.2, we discuss the DRL-based approaches in the recent CSMA literature, which serve as the second benchmark in our experiments later in Section 7.

3.1 Benchmark 1: Model-Based Approaches

The traditional model-based approaches (e.g., [1], [28], [5]) have a key assumption that the node density is fixed and known. To determine the optimal contention window thresholds, they derive the exact formulation of throughput based on a Markov model. In particular, [1] constructs a two-dimensional Markov chain (s_t, b_t) to model each node's state transition, where s_t denotes the collision number since the last successful transmission and b_t denotes the current backoff timer.

To derive the stationary probability τ that a node transmits in a generic time slot, [1] makes another assumption that at each transmission attempt, each packet collides with constant and independent probability p, where $p = 1 - (1-\tau)^{N-1}$ since each of the N nodes holds an independent transmission probability τ. Based on this probability p, the transition probabilities among the states can be obtained. Together with the stationary conditions, [1] obtains that τ and p are the solution to

$$\tau = \frac{2}{(1-p)\sum_{k=0}^{K-1} p^k W_k + p^K W_K + 1}, \; p = 1 - (1-\tau)^{N-1}. \quad (2)$$

Based on the transmission probability τ and the known node density N, [1] formulates throughput $U(\tau)$ as a function of τ:

$$U(\tau) = \frac{N\tau(1-\tau)^{N-1} T_P}{(1-\tau)^N T_\sigma + N\tau(1-\tau)^{N-1}(T_s - T_c) + (1-(1-\tau)^N) T_c}, \quad (3)$$

where T_P is the packet payload time, T_σ is the length of an empty slot time, T_s is the average time that the channel is sensed busy because of a successful transmission, and T_c is the average time the channel is sensed busy during a collision.

To maximize the throughput $U(\tau)$ in (3), [1] optimizes the contention window thresholds $\{W_k\}_{k=0}^K$ in (2) for reaching the optimal packet transmission probability τ^* given the knowledge of the node density N. Unfortunately, we have the following under a dynamic environment with an unknown node density.

Theorem 3.1. *Suppose that the unknown node density $N \leq \bar{N}$, and each contention window threshold $W_k \leq \bar{W}$ with $\bar{W} \geq 2\bar{N} - 1$. In a dynamic channel environment with an unknown node density N, the model-based approach with inaccurate estimation $\hat{N}$ of node density N leads to a large throughput loss*

$$U(\tau(N)) - U(\tau(\hat{N})) \geq \Theta\left(\frac{T_\sigma}{\bar{N}K^2\bar{W}^3 T_c^2}\right)|N - \hat{N}|.$$

Proof. See Appendix A. □

Theorem 3.1 indicates that the model-based approach leads to a certain throughput loss due to the inaccurate estimation of the node density. In highly dynamic channel environments, the gap between the actual node density N and the estimated $\hat{N}$ is large, leading to a large throughput loss. This motivates us to further develop an approach that can well adapt to unknown node densities.

3.2 Benchmark 2: DRL-based Approaches

In the recent CSMA literature (e.g., [3], [29], [30]), there are DRL-based approaches focusing on an unknown node density for optimizing the contention window thresholds. In particular, both [29] and [30] follow BEB to set the contention window threshold $W_k = 2^k W_0$ for each current collision number $k \in [K]$. The objective is to determine the best initial contention window threshold $W_0 = 2^{4+a}$, where a is chosen from a limited set $\{0, \cdots, 6\}$. They formulate (partially observable) Markov decision processes to learn the best action to choose W_0 by maximizing the Q functions. Note that in a dynamic channel environment where the node density keeps changing, models trained in one channel environment do not generalize well to other dynamic environments, which still lead to poor throughput with a large loss similar to Theorem 3.1.

Remark. DRL-based approaches in the recent CSMA literature focus on fixed channel environment parameters like node density. Thus, models trained in one channel environment do not generalize well to other dynamic environments, leading to a degraded throughput.

Later in Section 7.2, we run experiments to show that benchmark 2 of DRL approaches perform poorly if the node density keeps changing. According to the above analysis in Section 3, we are well motivated to propose a novel and efficient transformer-based ICL optimizer in Section 4. We use a transformer to perform ICL to predict the optimal contention window threshold given the current collision parameters, which only involves a few collision-threshold examples and does not need the unknown and varying node density.

4 Transformer-Based ICL for Optimizing CSMA

As shown in Figure 1, the construction of our transformer-based ICL for optimizing CSMA contains four steps: ICL data collection, prompt construction, embedding, and transformer training. In the following, we detail our design of each step.

4.1 Step I: ICL Data Collection

We define $x \in \mathbb{R}^d$ as the feature vector containing a set of d-dimensional collision parameters (e.g., the current collision number k, payload transmission time T_P, successful transmission time T_s, collision transmission time T_c, etc.). In practice, due to propagation delay variations, hardware clock inaccuracies, and processing jitter, these data related to packet transmission are noisy to fluctuate from time to time. Since the space of collision parameters is finite, we summarize all possible feature vectors x_i into a collision set $\mathcal{X} := \{x_i \in \mathbb{R}^d | i \in [I]\}$, where $\|x_i - x_{i'}\| = \Theta(\Delta)$ with feature vector gap $\Delta = \Theta(1)$ for any $i \neq i'$.

In a dynamic channel environment, since the node density changes dynamically over time, we use $\mathcal{S} \in \mathbb{R}^{d_s}$ to summarize all the possible node density environments. To ensure effective transformer training and adaptation to the dynamic node density environments, the system must sample multiple data pairs under different node densities. Consequently, for each node density environment $s \in \mathcal{S}$, we collect M prior data points for transformer training, where we let x_m^s denote each parameter for $m \in [M]$. We suppose x_m^s is a noisy version of some $x_i \in \mathcal{X}$, satisfying $\|x_m^s - x_i\| = O(\Delta)$. For simplicity, we assume $x_m^s = x_k$ in the later analysis, such that each x_m^s is randomly sampled from $\mathcal{X}$ with probability $\Theta(\frac{1}{K})$. For each

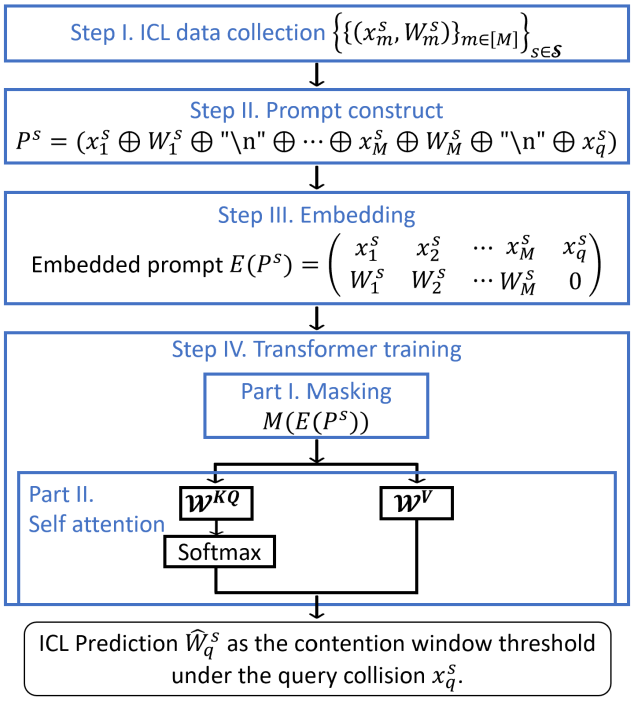

Figure 1: The overview of our transformer-based ICL optimizer in four steps: ICL data collection, prompt construction, embedding, and transformer training. First, the system collects M prior data points $\{(x_m^s, W_m^s)_{m\in[M]}\}$ and a query x_q^s as a new input with an unknown W_q^s in node density case $s \in \mathcal{S}$. Secondly, the system constructs prompt P^s using data points $\{(x_m^s, W_m^s)_{m\in[M]}\}$ and the query pair $(x_q^s, 0)$ for each s. Thirdly, each prompt P^s is embedded as $E(P^s)$, which is further masked as $M(E(P^s))$ to prevent the query input from attending to itself in the transformer training. Finally, $M(E(P^s))$ is multiplied by the $\mathcal{W}^{KQ}$ and $\mathcal{W}^V$ matrices with the self-attention mechanism to obtain the final prediction $\hat{W}_q^s$ in (8) as the contention window threshold under the query collision x_q^s. Our goal is to determine a near-optimal prediction $\hat{W}_q^s$ for the new query collision x_q^s based on the prior data points $\{(x_m^s, W_m^s)_{m\in[M]}\}$.

x_m^s of node density environment s, we let W_m^s denote its corresponding optimal contention window threshold under the current mapping function, defined as $f^s : \mathcal{X} \to \mathbb{N}$. In other words, we have $W_m^s = f^s(x_m^s)$. Note that the mapping functions f^s are unknown and vary with node density s, requiring us to learn them. For ease of exposition, we make the following assumptions about these mapping functions, which are more general than the transformer-based ICL literature (e.g., [32], [13], [17]).

Assumption 1. *Each mapping function f^s is independently drawn from a distribution $\mathcal{D}_f$ of the set $\mathcal{F} = \{f^s : \mathcal{X} \to \mathbb{N} | |f^s(x) - f^s(x')| \geq L\|x - x'\|, \forall \|x - x'\| \leq \delta_0\}$, where $L > 0$ and $\delta_0 = \Theta(1)$.*

There are many typical functions satisfying Assumption 1, such as linear, exponential, and scaled norm functions. In Section 7,

we relax Assumption 1 in experiments to verify the converged throughput loss under our transformer-based ICL optimizer.

For each feature vector x_m^s, we need to obtain its corresponding optimal W_m^s later for prompt construction and transformer training. We use a branch-and-bound algorithm (e.g., [22]) to solve the optimal probability τ^* that maximizes throughput U in (3) under a node density. Given τ^*, we then apply mixed-integer linear programming techniques (MILP), such as golden section search [4] and parabolic interpolation [12], to solve for the optimal contention window thresholds $\{W_m^s\}_{m=1}^M$ in (2). After successfully forming M data pairs $\{\{(x_m^s, W_m^s)\}_{m\in[M]}\}_{s\in\mathcal{S}}$ as prior training data, we sample a query x_q^s as a new input with an unknown W_q^s to be decided by our ICL optimizer. In Section 6, we relax this assumption of known optimal W_m^s and analyze the impact of erroneous data collection about contention window thresholds.

4.2 Steps II & III: Prompt Construction and Embedding

After obtaining M data pairs $\{(x_m^s, W_m^s)\}_{m\in[M]}$ and the new query x_q^s by Step I, the system then constructs each prompt P^s as follows:

$$P^s = x_1^s \oplus W_1^s \oplus \text{“\textbackslash n”} \oplus \cdots \oplus x_M^s \oplus W_M^s \oplus \text{“\textbackslash n”} \oplus x_q^s, s \in \mathcal{S}, \quad (4)$$

where $\oplus$ denotes the string concatenation operator, “\n” denotes a special delimiter token to distinguish data input. The last term x_q^s serves as the query of current collision parameter vector for predicting the optimal contention window threshold W_q^s, referred to as the query token. Since all examples within a prompt P^s correspond to the same node density, each collision pair of $\{(x_m^s, W_m^s)\}_{m\in[M]}$ and the query pair (x_q, W_q^s) follow the same mapping function $f^s \in \mathcal{F}$, satisfying $W_m^s = f^s(x_m^s)$ for any $m \in [M]$ and $W_q^s = f^s(x_q^s)$.

Given each prompt P^s in (4), we follow a natural token embedding in the ICL literature (e.g., [13]) to construct each column $m \in [M]$ as $\begin{pmatrix} x_m^s \\ W_m^s \end{pmatrix}$ and the last column as $\begin{pmatrix} x_q^s \\ 0 \end{pmatrix}$. Consequently, we obtain the embedding matrix of each P^s as follows:

$$E(P^s) = \begin{pmatrix} x_1^s & x_2^s & \cdots & x_M^s & x_q^s \\ W_1^s & W_2^s & \cdots & W_M^s & 0 \end{pmatrix} \in \mathbb{R}^{(d+1)\times(M+1)}. \quad (5)$$

Next, we use this embedding $E(P^s)$ to train the transformer.

4.3 Step IV: Transformer Training

Similar to the existing ICL literature (e.g., [13], [32]), we consider a simple but fundamental one-layer transformer, which contains a masking part and a self-attention part as in Figure 1. While simple, this facilitates our later theoretical analysis in Section 5, which demonstrates its effectiveness in optimizing CSMA and shows enough advantages over the benchmarks of Section 3.

To improve training efficiency, the system applies a masking operation to each embedding matrix $E(P^s)$ in (5), producing $M(E(P^s))$, which removes the last column to prevent the query input from attending to itself during training. The transformer then performs self-attention on the masked embedding $M(E(P^s))$. Below, we define the self-attention mechanism used in the transformer.

Definition 4.1. A self-attention (SA) layer in the single-head case consists a key matrix $\mathcal{W}^{\text{Key}} \in \mathbb{R}^{(d+1)\times(d+1)}$, a query matrix $\mathcal{W}^Q \in \mathbb{R}^{(d+1)\times(d+1)}$, and a value matrix $\mathcal{W}^V \in \mathbb{R}^{(d+1)\times(d+1)}$. Given an embedding E of a prompt P, the self-attention mechanism outputs

$$F_{SA}\big(E; \mathcal{W}^{\text{Key}}, \mathcal{W}^Q, \mathcal{W}^V\big) = \mathcal{W}^V E \cdot \text{softmax}\Big(\big(\mathcal{W}^{\text{Key}}E\big)^\top \mathcal{W}^Q E\Big),$$

where the softmax function is $\text{softmax}(z_i) = \frac{\exp(z_i)}{\sum_j \exp(z_j)}$ with z_i denoting the i-th element of vector $\boldsymbol{z}$.

We normalize the value matrix $\mathcal{W}^V$ to represent equal contribution from each collision-threshold pair in a prompt. Further, we consolidate the query and key matrices into one matrix as $\mathcal{W}^{KQ} \in \mathbb{R}^{d\times d}$ in the following forms:

$$\mathcal{W}^V = \begin{pmatrix} 0_{d\times d} & 0_d \\ 0_d^\top & 1 \end{pmatrix}, \quad \mathcal{W}^{KQ} = \begin{pmatrix} Q & 0_d \\ 0_d^\top & 0 \end{pmatrix}. \quad (6)$$

Note that the consolidation operation on the $\mathcal{W}^{KQ}$ matrix does not change the softmax input in F_{SA}. For ease of exposition, we use the notation $\theta = (1, Q)$ to represent all the transformer parameters for simplifying the transformer training analysis later in Section 5. Next, we are ready to give the self-attention mechanism in the parameter θ as follows:

$$F_{\text{SA}}(E(P^s);\theta) = M(E^W(P^s)) \cdot \text{softmax}\left(M(E^x(P^s))^\top Q E^x(P^s)\right), \quad (7)$$

where $E^x(P^s)$ and $E^W(P^s)$ denote the first d rows and the last row of $E(P^s)$, respectively. The ICL prediction for the query collision x_q^s is the last entry of F_{SA} as follows:

$$\hat{W}_q^s = \hat{W}_q^s(E(P^s);\theta) = \left[F_{\text{SA}}(E(P^s);\theta)\right]_{(M+1)}. \quad (8)$$

To train the transformer model over different node densities, we aim to minimize the following squared loss of the prediction error:

$$\mathcal{L}(\theta) = \mathbb{E}_{f^s\in\mathcal{F}}\left[\left(\hat{W}_q^s - W_q^s\right)^2\right], \quad (9)$$

where W_q^s is the optimal contention window threshold of x_q^s under f^s derived in Step I.

Observing $F_{\text{SA}}(E(P^s);\theta)$ in (7), we find that the non-linear softmax function couples the attention weights across all input tokens, making the training loss objective $\mathcal{L}(\theta)$ in (9) highly non-convex and interdependent. Although we have reduced the parameter space to $\theta = (1, Q)$, it is still difficult to explicitly solve the closed-form θ^* for minimizing $\mathcal{L}(\theta)$ in (9) with standard techniques in optimization theory. Consequently, we aim to propose a simple algorithm to efficiently train the transformer with a convergence guarantee after a limited number of training steps. To achieve this goal, we employ the gradient descent algorithm to optimize the non-convex and high-dimensional loss functions, which offers an efficient and scalable way to deal with our highly non-convex objective and is widely adopted in the machine learning literature (e.g., [13], [17], [32]). We then summarize details in Algorithm 1.

Although Algorithm 1 provides an efficient way to train the transformer parameters, it remains unclear how the parameter θ evolves during the training step, how long the training procedure takes at most, and whether the convergence of the final parameter θ^* can theoretically be guaranteed or not. We then make a comprehensive theoretical analysis in the next section.

Compared with the transformer-based ICL literature (e.g., [13], [32]), here the considered mapping function f^s from the collision parameters to the contention window threshold is more complex

Algorithm 1 Gradient descent for transformer training in Step IV of our transformer-based ICL optimizer

Input: Training loss objective $\mathcal{L}(\theta)$ in (9), transformer parameter $\theta = (1, Q)$, maximum training round number $\bar{T}$, step size η, precision error ϵ.
Output: Trained transformer parameter θ^*.
1: **Initialization**: $Q^{(0)} \leftarrow \mathbf{0}_{d\times d}$, $\theta^{(0)} \leftarrow (1, Q^{(0)})$.
2: Update the transformer parameter in a gradient descent way:
$$\theta^{(t+1)} = \theta^{(t)} - \eta \cdot \nabla_\theta \mathcal{L}(\theta^{(t)}).$$
3: **if** $\exists t < \bar{T}$ such that $||\theta^{(t+1)} - \theta^{(t)}||_2 \le \epsilon$ **then**
4: Record $\theta^* \leftarrow \theta^{(t+1)}$.
5: **else**
6: Record $\theta^* \leftarrow \theta^{(\bar{T})}$.

and no longer linear. Further, contention window thresholds are defined on a set of integers instead of a simple binary set, making it even more challenging to analyze the convergence performance.

5 Performance Analysis of Transformer-Based ICL for Optimizing CSMA

In this section, we aim to prove Algorithm 1's convergence and a limited throughput loss from the optimum. We need to build the connection between any collision example in the prompt and the query collision for a good transformer training and ICL prediction. To achieve this, we first define attention scores to measure how much the new query token x_q^s of P^s in (5) attends or relates to other input collision parameter vectors $\{x_m^s\}_{m\in[M]}$ during the transformer training process as in part II of Figure 1.

Definition 5.1. Given a prompt P^s in (4) and its corresponding embedding $E(P^s)$ in (5), we define the attention score at time t for the self-attention mechanism F_{SA} in (7) with parameter $\theta^{(t)}$ as follows.

a) Given $m \in [M]$, the attention score for the m-th collision token x_m is
$$\begin{aligned}\text{attn}_m^{(t)}(\theta^{(t)}; E^s(P^s)) :=& \Big[\text{softmax}\left(M(E^x(P^s))^\top Q E^x(P^s)\right)\Big]_m \\ =& \frac{\exp\Big(M\big(E^x(P^s)\big)^\top Q\big(\theta^{(t)}\big)\Big)}{\sum_{j\in[M]}\exp\Big(M\big(E_j^x(P^s)\big)^\top Q\big(\theta^{(t)}\big)\Big)}.\end{aligned}$$
b) For $k \in [K]$, define $\mathcal{X}_k^s(P^s) \subset [M]$ as the index set of collision input such that $x_m^s = x_k$ for $m \in \mathcal{X}_k^s(P^s)$. Then the attention score for the k-th collision token is given by
$$\text{Attn}_k^{(t)}(\theta^{(t)}; E^s(P^s)) := \sum_{m\in\mathcal{X}_k^s(P^s)} \text{attn}_m^{(t)}(\theta^{(t)}; E^s(P^s)).$$

For simplicity, we represent $\text{attn}_m^{(t)}(\theta^{(t)}; E^s(P^s))$ as $\text{attn}_m^{(t)}$ and $\text{Attn}_k^{(t)}(\theta^{(t)}; E^s(P^s))$ as $\text{Attn}_k^{(t)}$, respectively. We further rewrite $\mathcal{X}_k^s(P^s)$ as $\mathcal{X}_k^s$. Then the ICL prediction $\hat{W}_q^s$ in (8) for the current query token x_q^s can be rewritten as
$$\hat{W}_q^s = \sum_{m\in[M]} \text{attn}_m^{(t)} W_m^s = \sum_{k\in[K]} \text{Attn}_k^{(t)} f^s(x_k). \tag{10}$$

Based on the above definitions, we then establish the following criterion for Algorithm 1's convergence.

Proposition 5.2. *Given $\delta_0 = O(1)$, our Algorithm 1 achieves convergence if and only if, for any $t > T^*$ with query token $x_q^s = x_k$, the attention score with respect to x_k satisfies*
$$1 - Attn_k^{(t)} = O(\delta_0). \tag{11}$$

Proof. See Appendix B. □

Intuitively, in the convergence state, if the query token is $x_q^s = x_k$, the transformer correctly outputs the optimal contention window threshold $f^s(x_k)$ for x_q^s by assigning an attention score $\text{Attn}_k^{(t)}$ close to 1 in (10). Based on the convergence in attention score $\text{Attn}_k^{(t)}$ in Proposition 5.2, we now establish the convergence guarantee of our Algorithm 1.

Theorem 5.3. *If the number of input examples M in a prompt P^s in (4) and the maximum collision number K satisfy $M \ge poly(K)$, and the gap parameter L of mapping functions satisfies $L = O(\frac{1}{\Delta})$, then for any $\epsilon\in(0,1)$, applying Algorithm 1 to train the loss function $\mathcal{L}(\theta)$ in (9) ensures convergence. Specifically, with at most $T^* = \Theta(\frac{K\log(K\epsilon^{-1})}{\eta\delta_0^2 L^2\Delta^2})$ iterations, the prediction loss satisfies $\mathcal{L}\big(\theta^{(T^*)}\big)=O(\epsilon^2)$.*

Proof. See Appendix C. □

Theorem 5.3 states that with a sufficiently large number of data examples M, Algorithm 1 ensures convergence of the loss function L in (9) within a limited number of training steps, allowing the transformer to output a near-optimal $\hat{W}_q^s$ for x_q^s. However, as the maximum collision number K increases, each prompt needs to include more data examples to train, which delays the convergence. Conversely, the convergence time T^* in Theorem 5.3 decreases with both the function gap parameter L and feature vector gap Δ defined in Assumption 1 since a larger L or Δ results in a greater gradient, which accelerates the loss function convergence. Unlike DRL benchmarks, which require full retraining when the environment changes, our transformer-based ICL optimizer adapts efficiently by focusing on the iteration count needed for convergence, making it well-suited for dynamic node density environments.

To further check the throughput U in (1) of our transformer-based ICL approach, we define the throughput loss ΔU as the expected difference between the throughput U^* under the optimal contention window threshold W_q^s and $\hat{U}$ of the ICL prediction $\hat{W}_q^s$ for all the prompts:
$$\Delta U := U^* - \hat{U} = \mathbb{E}_{f^s\in\mathcal{F}}\big[U(W_q^s) - U(\hat{W}_q^s)\big]. \tag{12}$$

To bound the throughput loss ΔU in (12) given the training loss of our approach in Theorem 5.3, we further prove the Lipschitz continuity of U according to (2) and (3) in the following.

Lemma 5.4. *Given $N \le \bar{N}$, the throughput U in (3) is $\frac{T_P\bar{N}}{8T_\sigma}$-Lipschitz continuous in each contention window threshold W_k:*
$$|U(W_k) - U(\hat{W}_k)| \le \frac{T_P\bar{N}}{8T_\sigma}\cdot|W_k - \hat{W}_k|.$$

Proof. See Appendix D. □

According to Theorem 5.3 and Lemma 5.4, we are now ready to well bound the throughput loss ΔU in (12) explicitly.

Theorem 5.5. *The throughput loss ΔU in (12) of our transformer-based ICL approach from the optimum is upper bounded as follows:*

$$\Delta U \leq O\left(\frac{T_P \bar{N} \epsilon}{8 T_\sigma}\right).$$

Proof. See Appendix E. □

Compared with benchmark 1's large throughput loss in Theorem 3.1, Theorem 5.5 guarantees a limited throughput loss for our transformer-based ICL approach. Intuitively, as the maximum node density $\bar{N}$ increases, packet collision may occur more frequently and the system needs to carefully determine the contention window thresholds for reducing collisions to improve the throughput U. Accordingly, a small deviation from the optimum contention window thresholds can still lead to a large throughput loss ΔU, resulting in a larger bound $O(\frac{T_P \bar{N} \epsilon}{8 T_\sigma})$.

6 Robustness to the Erroneous Prompts

Recall that in Step I of data collection in Section 4.1, we assume that in each prompt P^s in (4), each contention window threshold W_m^s is optimal for the feature vector x_m^s, $m \in [M]$ and $s \in \mathcal{S}$. In practice, it can be difficult for the system to collect the optimal data for each collision case. In this section, we remove this assumption and allow each W_m^s to be erroneous for collision x_m^s for further evaluation.

6.1 Extended System Model of the Erroneous Prompts

The analysis of our transformer-based ICL optimizer in Sections 4 and 5 requires each collision-threshold pair (x_m^s, W_m^s) to follow the same mapping f^s in each prompt P^s, $s \in \mathcal{S}$, which no longer holds if W_m^s is erroneous for x_m^s. We are then motivated to develop our analysis based on a general LLM, which is pre-trained to learn an arbitrary $f \in \mathcal{D}$ between the collision parameters and the corresponding contention window threshold under another node density. Denote $\mathbb{P}_f(y|x)$ as the probability that the LLM generates a contention window threshold W given an input collision x under the mapping f, where x, W are taken from a set Ω. The mapping f can be efficiently learned, i.e., the LLM is well pre-trained with a distribution $\mathbb{P}_\theta(\cdot|\cdot)$ such that for any mapping $f \in \mathcal{F}$, we have

$$\max_{x,W \in \Omega} |\mathbb{P}_f(W|x) - \mathbb{P}_\theta(W|x)| < \Delta_{pre}.$$

The LLM is supposed to be well pre-trained with a sufficiently small gap Δ_{pre} close to 0. We consider a challenging case that the mapping $f^* \in \mathcal{D}^*$ in the prompt P to learn under an unknown target node density is not the same as f for pre-training of another node density in general. In practice, the prompt P containing collision-threshold examples does not resemble inputs that the LLM has been pre-trained on. Thus, either two consecutive example strings $s_1 = x_1 \oplus W_1 \oplus \cdots \oplus x_m \oplus W_m$ and $s_2 = x_{m+1} \oplus W_{m+1}$ from the prompt set Ω^* with $m \leq M-1$ are approximately independent according to the pretraining mapping, modeled as:

$$\alpha \mathbb{P}_f(s_2|s_1 \oplus \text{“\textbackslash n”}) \leq \mathbb{P}_f(s_2) \leq \frac{1}{\alpha} \mathbb{P}_f(s_2|s_1 \oplus \text{“\textbackslash n”}),\ \alpha \in (0,1].$$

To avoid zero likelihood due to the unnatural concatenation of collision-threshold examples in the prompt P, we assume that there exists a constant $\beta > 0$ such that for any token t in the prompt set Ω^*, any token t' in the pretaining set Ω and any mapping $f \in \mathcal{F}$, we have $\mathbb{P}_f(t|t') > \beta$. Finally, we consider that there is a positive probability that the pretraining distribution $\mathcal{D}$ generates the mapping f^* of the prompt, i.e., $Pr(f^*|\mathcal{D}) \geq \gamma > 0$.

To investigate the performance of the LLM's ICL, we model the zero-one loss of the in-context predictor as follows (e.g., [27]):

$$\mathcal{L} := \mathbb{E}_{x,W \sim \mathcal{D}^*}[\mathbf{1}(\arg\max_{W'} \mathbb{P}_\theta(P \oplus W') \neq W)]. \tag{13}$$

Note that $\mathcal{L}$ in (13) is similar to that in (9) to capture the ICL prediction loss to the optimal contention window thresholds.

6.2 Robustness Analysis to Erroneous Prompts

Denote ℓ as the length of each collision input x. In the following, we first introduce an important lemma for further analysis.

Lemma 6.1. *Suppose that the minimum KL-divergence between our ICL mapping f and the ground-truth f^* satisfies $\min_f KL(\mathbb{P}_f, \mathbb{P}_{f^*}) > -8\ln(\alpha\beta)$. If the number of in-context examples M is long enough as*

$$M \geq \max\left\{\frac{-(\ln q)(16\ell^2)(\ln^2 \beta)}{KL^2(\mathbb{P}_f, \mathbb{P}_{f^*})}, \frac{-2\ln(\frac{\mathbb{P}_{\mathcal{D}^*}(W|x) - \mathbb{P}_{\mathcal{D}^*}(\hat{W}|x)}{5\alpha^{-2}\beta^{-T}\gamma^{-1}})}{\min_f KL(\mathbb{P}_f, \mathbb{P}_{f^*}) + 8\ln(\alpha\beta)}\right\},$$

$\mathbb{P}_{\mathcal{D}^*}(W|x) - \mathbb{P}_{\mathcal{D}^*}(\hat{W}|x) > 0$ and any $q \in (0,1)$ for every collision case x and two contention threshold candidates $W, \hat{W}$, we have

$$Pr\left(\frac{\mathbb{P}_{\mathcal{D}^*}(W|x) - \mathbb{P}_{\mathcal{D}^*}(\hat{W}|x)}{2} - \left(\mathbb{P}_{\mathcal{D}}(W|P) - \mathbb{P}_{\mathcal{D}}(\hat{W}|P)\right) < 1 - \alpha^2\right) \geq 1 - q. \tag{14}$$

Proof. See Appendix F. □

Lemma 6.1 shows that for any mapping distribution $\mathcal{D} \neq \mathcal{D}^*$, we can still guarantee that the margin difference between generating any two contention thresholds W and $\hat{W}$ given the prompt P under the pre-trained mapping distribution $\mathcal{D}$ is at least half of the margin difference under the ground-truth mapping distribution $\mathcal{D}^*$. In other words, the LLM is still able to distinguish the correct contention window threshold even if we change the input distribution by concatenating examples.

Based on Lemma 6.1, we are now ready to prove our main ICL result. Denote $\Delta_{\mathcal{D}^*}$ as the minimal difference between Bayes Optimal Classifier prediction and any mapping $f \in \mathcal{D}^*$. We have the following.

Theorem 6.2. *Given the margin $\Delta_{\mathcal{D}^*}$ of the prompt mapping distribution satisfying $\Delta_{\mathcal{D}^*} > 1 - \alpha^2$ and denote $c := 1 - \sqrt{\frac{1-\alpha^2}{\Delta_{\mathcal{D}^*}}} \in (0,1)$, the ICL prediction loss $\mathcal{L}$ in (13) satisfies*

$$\mathcal{L} \leq \frac{2\Delta_{pre}}{c(1-c)}.$$

Besides, the throughput loss ΔU due to an erroneous prompt is upper bounded as follows:

$$\Delta U \leq O\left(\frac{T_P \bar{N}}{8 T_\sigma}\left(\frac{2\Delta_{pre}}{c(1-c)}\right)^{\frac{1}{2}} \bar{W}\right).$$

Proof. See Appendix G. □

Note that if the pre-training error $\Delta_{pre} = 0$, the ICL prediction loss $\mathcal{L}$ in Theorem 6.2 also converges to 0, indicating our ICL approach's adaptiveness to any other node density. Since the contention window threshold can be erroneous in each collision case in any prompt, the upper bound in Theorem 6.2 is further constrained by the maximum contention window threshold $\bar{W}$ compared with Theorem 5.5 in the perfect prompt case.

7 Simulation Experiments

In Section 7.1, we introduce our experimental settings of the co-used channel. In Section 7.2, we introduce our experimental results in the training stage of four steps as illustrated in Figure 1. In Section 7.3, we introduce our experimental results in the testing stage with unknown and heavy node densities.

7.1 Experiment Settings

Following the CSMA literature (e.g., [1], [33]), we consider the same settings as the DCF in 802.11 protocols. We set a unit slot time $T_\sigma = 50\mu s$, the DIFS time $T_{\text{DIFS}} = 128\mu s$ for sensing the channel as idle, the short interframe space (SIFS) $T_{\text{SIFS}} = 28\mu s$ as the smallest waiting interval between the transmission of a data frame and its subsequent response (e.g., ACK), the propagation delay $T_\delta = 1\mu s$ as the time needed for informing the channel state change to all the contention nodes (i.e., from idle to busy or from busy to idle). Besides, we consider the channel bit rate as $1Mbit/s$ shared by N variable nodes, the acknowledging packet length as ACK = 240 bits, the packet header length (for the physical and MAC layers) as 400 bits, and the packet payload length as 8184 bits. According to the channel bit rate and length of packets, we obtain the time for acknowledging the contention nodes of a successful transmission as $T_{\text{ACK}} = 240\mu s$, the time for transmitting the packet header $T_H = 400\mu s$, and the time for transmitting the packet payload $T_P = 8184\mu s$. Note that the above settings are essential for obtaining the closed-form throughput U in (3) for ICL data collection and further comparisons.

According to DCF, a successful transmission period T_s begins with the transmission time of the packet header T_H, followed by the packet payload time T_P. After transmitting the packet, DCF requires a short interframe space (SIFS) T_{SIFS} before acknowledging the node of the successful transmission, which takes a time of T_{ACK}. The successful transmission period ends with a DIFS time T_{DIFS} and a propagation delay T_δ for other nodes to detect the channel idle. In summary, we write T_s as follows:

$$T_s = T_H + T_P + T_{\text{SIFS}} + T_\delta + T_{\text{ACK}} + T_{\text{DIFS}} + T_\delta = 8.982ms.$$

A collision period T_c also begins with the transmission time of the packet header T_H, followed by the packet payload time T_P. Nonetheless, after a collision occurs, each active node cannot receive an ACK and then stops any further transmission to backoff, which requires a DIFS time T_{DIFS} and a propagation delay T_δ for other nodes to detect the channel idle. We write T_c as follows:

$$T_c = T_H + T_P + T_{\text{DIFS}} + T_\delta = 8.783ms.$$

To simulate the dynamic environment of node density, we use prompts generated under low density environments to train the transformer parameters in Section 7.2. We then use the trained transformer to test under high unknown node density environments of even hundreds of nodes in Section 7.3.

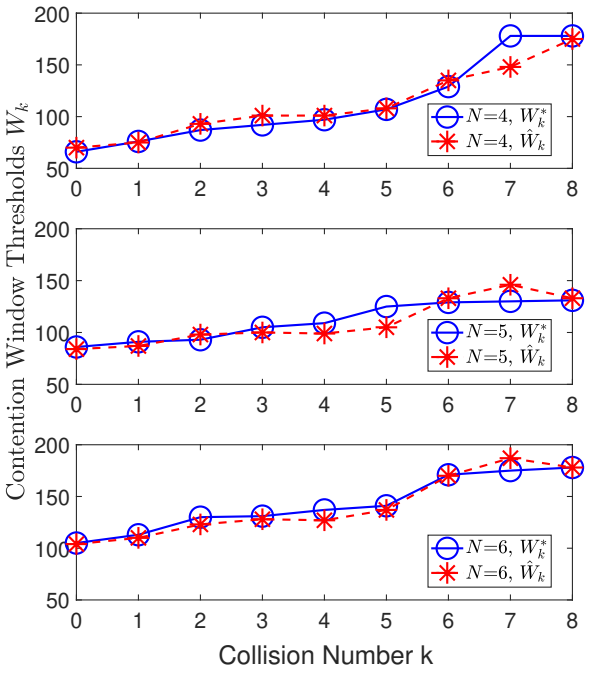

Figure 2: Contention window thresholds $\{W_k^*\}_{k=0}^K$ at the optimum and $\{\hat{W}_k\}_{k=0}^K$ of our transformer-based ICL approach versus the collision number k, respectively. Here we fix the maximum collision number $K = 8$ and change the node density $N \in \{4, 5, 6\}$ across the three subfigures as included in the training stage.

7.2 Experimental Results in the Training Stage

We consider node density environments $N \in \{2, 3, 4, 5, 6\}$ for ICL data collection. As in Step I of Figure 1, for each node density N, the system collects $M = 9$ data points $\{(x_k, W_k^*)\}_{k=0}^8$ to construct the prompt, where $x_k = (k, T_P, T_s, T_c)$, $k \in [K]$ denotes the current collision number and the maximum collision number is set as $K = 8$. Note that the node density N is not shown in any x_k. To obtain the contention window thresholds $\{W_k^*\}_{k=0}^8$ in a prompt with a node density N, the system first maximizes the throughput U in (3) for obtaining the optimal packet transmission probability τ^* (e.g., using the branch-and-bound algorithm). Then, it uses the τ^* to solve $\{W_k^*\}_{k=0}^8$ according to (2) by applying MILP algorithms like golden section search and parabolic interpolation.

As in Step II of Figure 1, we construct $S = 5$ prompts corresponding to 5 different node densities $N \in \{2, 3, 4, 5, 6\}$ for training the transformer parameters θ, where each prompt contains 8 example points (x_k, W_k^*) and a query collision case x_q^s. These prompts are then embedded as in Step III of Figure 1. Regarding Step IV of Figure 1 for transformer training, we set the step size of Algorithm 1 as $\eta = 0.05$. We consider $x_q^s = x_k$ to be the same for all 5 prompts and obtain our ICL prediction $\hat{W}_k = \hat{W}_q^s$ in (8) of each contention window threshold W_k^*, $k \in [K]$.

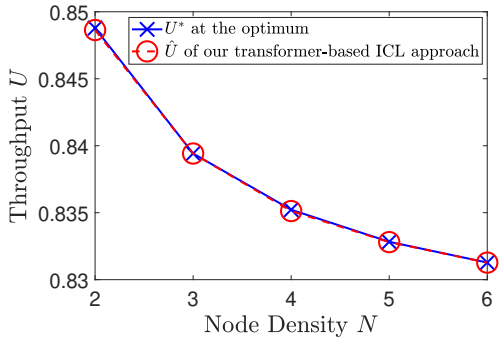

Figure 3: The throughput U^* at the optimum and $\hat{U}$ of our transformer-based ICL approach versus the node density N, respectively. Here we check for each node density $N \in \{2, 3, 4, 5, 6\}$ as included in the training stage.

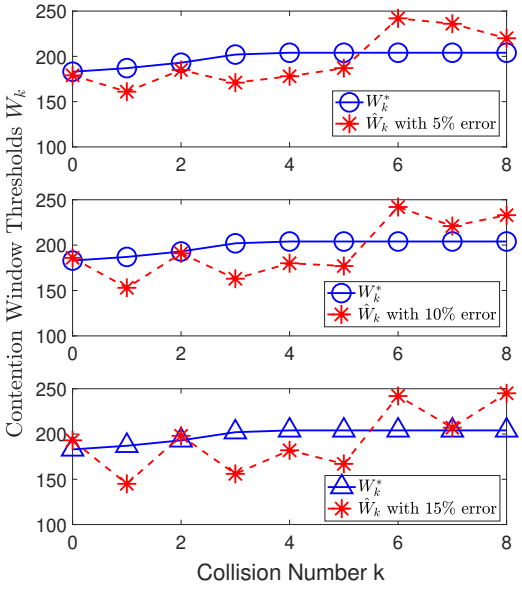

Figure 4: Contention window thresholds $\{W_k^*\}_{k=0}^K$ at the optimum and $\{\hat{W}_k\}_{k=0}^K$ of our transformer-based ICL approach with erroneous prompts versus the collision number k, respectively. Here we change the erroneous percentage $b \in \{5, 10, 15\}$ and test under the unknown node density $N = 10$.

Figure 2 plots contention window thresholds $\{W_k^*\}_{k=0}^K$ at the optimum and $\{\hat{W}_k\}_{k=0}^K$ of our transformer-based ICL approach versus the collision number k, respectively. It shows that our approach can approximate each optimal contention window threshold in the node density environment of the training stage, which is consistent with Theorem 5.3 in Section 5.

Figure 3 plots the throughput U^* at the optimum and $\hat{U}$ of our transformer-based ICL approach versus the node density N, respectively. It indicates that our approach can nearly achieve the optimal throughput in each node density N considered in the training stage, which is consistent with the limited throughput loss in Proposition 5.5 of Section 5.

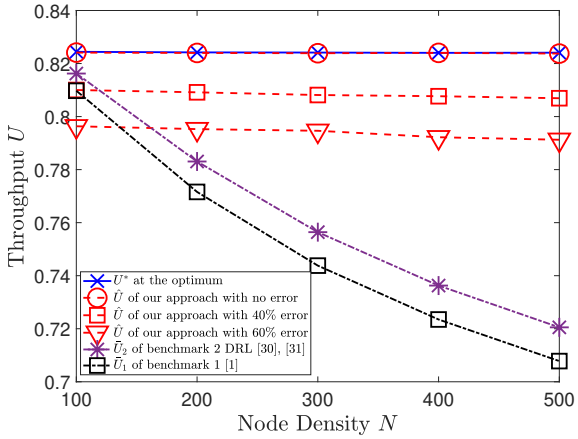

Figure 5: The throughput U^* at the optimum, $\hat{U}$ of our transformer-based ICL approach with no error, 40% error and 60% error, $\bar{U}_1$ of the benchmark 1 of BEB, and $\bar{U}_2$ of the benchmark 2 of DRL versus the node density N, respectively. Here we optimize benchmarks 1 and 2 under an approximated node density of $\hat{N} = 50$ and change the unknown node density $N \in [100, 500]$.

7.3 Experimental Results in the Testing Stage

In the following, we want to test the adaptiveness of our transformer-based ICL approach to dynamic node density environments. We consider a challenging case where each contention threshold $\tilde{W}_k$ is erroneous in each testing prompt as in Section 6. In particular, we define that a prompt is with b% error if each erroneous threshold $\tilde{W}_k$ is randomly realized from the set of $\{(1-100b)W_k^*, (1+100b)W_k^*\}$ regarding the ground-truth W_k^*, where $b \in (0, 100)$. We follow the same steps as in the training stage to construct prompts for testing.

Figure 4 plots contention window thresholds $\{W_k^*\}_{k=0}^K$ at the optimum and $\{\hat{W}_k\}_{k=0}^K$ of our transformer-based ICL approach with erroneous prompts versus the collision number k, respectively, where we consider a unknown node density N=10 beyond the training density set $\{2, \cdots, 6\}$. Though the data of the testing node density environment $N = 10$ does not appear in the training stage and the prompts are erroneous in the testing stage, our approach can still approximate the optimal contention window thresholds well for some certain collision cases and keep a small prediction error, which is consistent with Proposition 6.2 of Section 6.

Figure 5 plots the throughput U^* at the optimum, $\hat{U}$ of our transformer-based ICL approach with no error, 40% error and 60% error, $\bar{U}_2$ of the benchmark 2 of DRL, and $\bar{U}_1$ of the benchmark 1 of BEB versus the node density N even scales up to hundreds, respectively. It shows that our approach with no error still achieves the near-optimal throughput even under high and unknown node densities, which is consistent with Theorem 5.3 of Section 5. Further, Figure 5 indicates that the throughput loss of our approach with

erroneous prompts are still limited even under high and unknown node densities, which is consistent with Theorem 6.2 of Section 6.

Figure 5 also shows that our transformer-based ICL approach with no error always outperforms the benchmark schemes and our transformer-based ICL approach with large errors still outperforms either as long as the unknown node density N is larger than 200. The throughput difference between benchmark 1 of BEB and the optimum enlarges as the gap between estimated node density and the ground truth increases, which is consistent with Theorem 3.1 of Section 3.1.

8 Conclusion

In this paper, we study transformer-based ICL for optimizing CSMA. Our approach contains four steps: ICL data collection, prompt construction, embedding, and transformer training. We construct prompts as the input to the transformer by pre-collecting multiple points (of the current collision parameters and the corresponding contention window thresholds) and a query collision case. To train the transformer, we employ gradient descent to develop an efficient algorithm to analyze its learning convergence. Our analysis guarantees that within a limited number of steps, the transformer accurately predicts a near-optimal contention window threshold for any given query token, ensuring a converged throughput loss. We further extend to consider erroneous data input examples with erroneous contention window thresholds to the transformer. We prove that our transformer-based ICL optimizer still incurs limited ICL prediction and throughput losses to the optimum. Experimental results further demonstrate our transformer-based ICL optimizer's great advantage over benchmarks in well adapting to unknown and large node densities.

References

[1] Giuseppe Bianchi. 2000. Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE Journal on selected areas in communications* 18, 3 (2000), 535–547.

[2] Gordon Owusu Boateng, Hani Sami, Ahmed Alagha, Hanae Elmekki, Ahmad Hammoud, Rabeb Mizouni, Azzam Mourad, Hadi Otrok, Jamal Bentahar, Sami Muhaidat, et al. 2024. A Survey on Large Language Models for Communication, Network, and Service Management: Application Insights, Challenges, and Future Directions. *arXiv preprint arXiv:2412.19823* (2024).

[3] Xuelin Cao, Bo Yang, Kaining Wang, Xinghua Li, Zhiwen Yu, Chau Yuen, Yan Zhang, and Zhu Han. 2024. AI-empowered multiple access for 6G: A survey of spectrum sensing, protocol designs, and optimizations. *Proc. IEEE* (2024).

[4] Yen-Ching Chang. 2009. N-dimension golden section search: Its variants and limitations. In *2009 2nd International Conference on Biomedical Engineering and Informatics*. IEEE, 1–6.

[5] Nicola Cordeschi, Floriano De Rango, and Andrea Baiocchi. 2024. Optimal Back-Off Distribution for Maximum Weighted Throughput in CSMA. *IEEE/ACM Transactions on Networking* (2024).

[6] Lin Dai. 2022. A theoretical framework for random access: Stability regions and transmission control. *IEEE/ACM Transactions on Networking* 30, 5 (2022), 2173–2200.

[7] Lin Dai. 2024. A Theoretical Framework for Random Access: Effects of Carrier Sensing on Stability. *IEEE Transactions on Communications* (2024).

[8] Cailian Deng, Xuming Fang, Xiao Han, Xianbin Wang, Li Yan, Rong He, Yan Long, and Yuchen Guo. 2020. IEEE 802.11 be Wi-Fi 7: New challenges and opportunities. *IEEE Communications Surveys & Tutorials* 22, 4 (2020), 2136–2166.

[9] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Baobao Chang, et al. 2024. A Survey on In-context Learning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. 1107–1128.

[10] Yayu Gao, Shuangfeng Fang, Xiangchen Song, and Lin Dai. 2022. When Aloha and CSMA coexist: Modeling, fairness, and throughput optimization. *IEEE Transactions on Wireless Communications* 21, 10 (2022), 8163–8178.

[11] Zhiwu Guo, Ming Li, and Marwan Krunz. 2023. Exploiting successive interference cancellation for spectrum sharing over unlicensed bands. *IEEE Transactions on Mobile Computing* 23, 3 (2023), 2438–2455.

[12] Michael T Heath and Scientific Computing. 2002. An Introductory Survey. *McGrawHill-* (2002).

[13] Yu Huang, Yuan Cheng, and Yingbin Liang. 2024. In-context convergence of transformers. In *Proceedings of the 41st International Conference on Machine Learning*. 19660–19722.

[14] Evgeny Khorov, Anton Kiryanov, Andrey Lyakhov, and Giuseppe Bianchi. 2018. A tutorial on IEEE 802.11 ax high efficiency WLANs. *IEEE Communications Surveys & Tutorials* 21, 1 (2018), 197–216.

[15] Woongsup Lee and Jeonghun Park. 2024. LLM-empowered resource allocation in wireless communications systems. *arXiv preprint arXiv:2408.02944* (2024).

[16] Haiyuan Li, Hari Madhukumar, Peizheng Li, Yiran Teng, Shuangyi Yan, and Dimitra Simeonidou. 2024. From Hype to Reality: The Road Ahead of Deploying DRL in 6G Networks. *arXiv preprint arXiv:2410.23086* (2024).

[17] Hongkang Li, Meng Wang, Songtao Lu, Xiaodong Cui, and Pin-Yu Chen. 2024. How Do Nonlinear Transformers Learn and Generalize in In-Context Learning?. In *International Conference on Machine Learning*. PMLR, 28734–28783.

[18] Jiayu Liu, Zhenya Huang, Chaokun Wang, Xunpeng Huang, Chengxiang Zhai, and Enhong Chen. 2024. What makes in-context learning effective for mathematical reasoning: A theoretical analysis. *arXiv preprint arXiv:2412.12157* (2024).

[19] Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. [n. d.]. MathVista: Evaluating Mathematical Reasoning of Foundation Models in Visual Contexts. In *The Twelfth International Conference on Learning Representations*.

[20] Xiao Mao, Guohua Wu, Mingfeng Fan, Zhiguang Cao, and Witold Pedrycz. 2024. DL-DRL: A double-level deep reinforcement learning approach for large-scale task scheduling of multi-UAV. *IEEE Transactions on Automation Science and Engineering* 22 (2024), 1028–1044.

[21] Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the Role of Demonstrations: What Makes In-Context Learning Work?. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*. 11048–11064.

[22] David R Morrison, Sheldon H Jacobson, Jason J Sauppe, and Edward C Sewell. 2016. Branch-and-bound algorithms: A survey of recent advances in searching, branching, and pruning. *Discrete Optimization* 19 (2016), 79–102.

[23] Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2022. Learning To Retrieve Prompts for In-Context Learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2655–2671.

[24] Nurullah Sevim, Mostafa Ibrahim, and Sabit Ekin. 2024. Large language models (LLMs) assisted wireless network deployment in urban settings. In *2024 IEEE 100th Vehicular Technology Conference (VTC2024-Fall)*. IEEE, 1–7.

[25] Rahul Singh and PR Kumar. 2021. Adaptive CSMA for decentralized scheduling of multi-hop networks with end-to-end deadline constraints. *IEEE/ACM Transactions on Networking* 29, 3 (2021), 1224–1237.

[26] Zihang Song, Osvaldo Simeone, and Bipin Rajendran. 2024. Neuromorphic In-Context Learning for energy-efficient MIMO symbol detection. In *2024 IEEE 25th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*. IEEE, 1–5.

[27] Noam Wies, Yoav Levine, and Amnon Shashua. 2023. The learnability of in-context learning. *Advances in Neural Information Processing Systems* 36 (2023), 36637–36651.

[28] Pui King Wong, Dongjie Yin, and Tony T Lee. 2011. Analysis of non-persistent CSMA protocols with exponential backoff scheduling. *IEEE Transactions on Communications* 59, 8 (2011), 2206–2214.

[29] Witold Wydmański and Szymon Szott. 2021. Contention window optimization in IEEE 802.11 ax networks with deep reinforcement learning. In *2021 IEEE wireless communications and networking conference (WCNC)*. IEEE, 1–6.

[30] Rong Yan, Mingjun Du, Xiao-Ping Zhang, and Yuhan Dong. 2024. Deep Reinforcement Learning Based Contention Window Optimization for IEEE 802.11 bn. In *2024 IEEE 99th Vehicular Technology Conference (VTC2024-Spring)*. IEEE, 1–5.

[31] Han Zhang, Akram Bin Sediq, Ali Afana, and Melike Erol-Kantarci. 2024. Large language models in wireless application design: In-context learning-enhanced automatic network intrusion detection. *arXiv preprint arXiv:2405.11002* (2024).

[32] Ruiqi Zhang, Spencer Frei, and Peter L Bartlett. 2024. Trained transformers learn linear models in-context. *Journal of Machine Learning Research* 25, 49 (2024), 1–55.

[33] Lei Zheng, Minming Ni, Lin Cai, Jianping Pan, Chittabrata Ghosh, and Klaus Doppler. 2014. Performance analysis of group-synchronized DCF for dense IEEE 802.11 networks. *IEEE Transactions on Wireless Communications* 13, 11 (2014), 6180–6192.

[34] Hao Zhou, Chengming Hu, Dun Yuan, Ye Yuan, Di Wu, Xue Liu, and Charlie Zhang. 2024. Large language model (llm)-enabled in-context learning for wireless network optimization: A case study of power control. *arXiv preprint arXiv:2408.00214* (2024).

A Proof of Theorem 3.1

Before proving the theorem, we introduce a useful lemma, which is obtained by checking the first derivative of throughput U in (3).

Lemma A.1. *The optimal solution τ^* to maximize $U(\tau)$ in (3) satisfies $\tau^* < 1/N$.*

To prove the lower bound in Theorem 3.1, we first prove that there exists a $C_1 > 0$ such that $|\tau(N) - \tau(\hat{N})| \geq C_1|N - \hat{N}|$. Then, we prove that there exists a $C_2 > 0$ such that $|U(\tau(N)) - U(\tau(\hat{N}))| \geq C_2|\tau(N) - \tau(\hat{N})|$.

To find the exact C_1, we define

$$D(N,\tau) := (1-(1-\tau)^{N-1})\sum_{k=0}^{K-1}(1-(1-\tau)^{N-1})^k W_k + (1-(1-\tau)^{N-1})^K W_K + 1.$$

According to (2), we have $\tau = \frac{2}{D}$. Differentiating $\tau = \frac{2}{D}$ with respect to N, we have

$$\frac{\partial \tau}{\partial N} = -\frac{\tau\left(\frac{\partial D}{\partial N} + \frac{\partial D}{\partial \tau}\frac{\partial \tau}{\partial N}\right)}{D}.$$

After rewriting the above equality, we obtain that

$$\left|\frac{\partial \tau}{\partial N}\right| = \frac{|\tau \frac{\partial D}{\partial N}|}{|D + \tau \frac{\partial D}{\partial \tau}|}.$$

According to the Mean Value Theorem, there exists a $\tau_0 \in (0,1)$ such that

$$|\tau(N) - \tau(\hat{N})| = \left|\frac{\partial \tau}{\partial N}\Big|_{\tau=\tau_0}\right||N - \hat{N}|.$$

Therefore, to prove that $|\tau(N) - \tau(\hat{N})| \geq C_1|N - \hat{N}|$, it is enough to prove that $|\frac{\partial \tau}{\partial N}| \geq C_1$. This motivates us to find a lower bound of the numerator of $|\frac{\partial \tau}{\partial N}|$ and an upper bound of the denominator of $|\frac{\partial \tau}{\partial N}|$. The partial derivative of $D(N,\tau)$ on N is:

$$\frac{\partial D}{\partial N} = \frac{\partial D}{\partial p}\frac{\partial p}{\partial N} = \left(\sum_{k=0}^{K-1}(k+1)p^k W_k + Kp^{K-1}W_K\right)\left(-(1-\tau)^{N-1}\ln(1-\tau)\right) \geq (1-\tau)^{N-1}\tau, \tag{15}$$

where the inequality holds due to

$$\sum_{k=0}^{K-1}(k+1)p^k W_k + Kp^{K-1}W_K \geq \sum_{k=0}^{K-1}(k+1)p^k W_k \geq \sum_{k=0}^{K-1}(k+1)p^k \geq 1$$

and $-\ln(1-\tau) \geq \tau$ for $\tau \in (0,1)$. The partial derivative of $D(N,\tau)$ on τ is:

$$\frac{\partial D}{\partial \tau} = \frac{\partial D}{\partial p}\frac{\partial p}{\partial \tau} = \left(\sum_{k=0}^{K-1}(k+1)p^k W_k + Kp^{K-1}W_K\right)(N-1)(1-\tau)^{N-2} \leq \bar{W}(N-1)(1-\tau)^{N-2}\left(\frac{K(K+1)}{2} + K\right) \tag{16}$$

due to $W_k \leq \bar{W}$ and $p \leq 1$ for $k \in [K]$. Based on (15) and (16), we obtain a lower bound of $|\frac{\partial \tau}{\partial N}|$ as follows:

$$\left|\frac{\partial \tau}{\partial N}\right| \geq \frac{\tau(1-\tau)^{N-1}\tau}{\frac{2}{\tau} + \tau\bar{W}(N-1)(1-\tau)^{N-2}\left(\frac{K(K+1)}{2} + K\right)}. \tag{17}$$

Since $W_k \leq \bar{W}$, we have $\tau = \frac{2}{D} \geq \frac{2}{\bar{W}+1}$. Together with $\tau \leq \frac{1}{N} \leq \frac{1}{2}$, $(1-\tau)^{N-1} \geq e^{-1}$ for $\tau \leq \frac{1}{N}$ and $(1-\tau)^{N-2} \leq 1$, we further bound $\left|\frac{\partial \tau}{\partial N}\right|$ according to (17) as follows:

$$\left|\frac{\partial \tau}{\partial N}\right| \geq \frac{e^{-1}\frac{4}{(\bar{W}+1)^2}}{\frac{4}{\bar{W}+1} + \frac{1}{2}\bar{W}(\bar{N}-1)\left(\frac{K(K+1)}{2} + K\right)} = \Theta\left(\frac{1}{\bar{N}\bar{W}^3K^2}\right) = C_1.$$

To find the exact C_2, similar to the analysis of $\tau(n)$, we can check that $U(\tau)$ in (3) satisfies $|U'(\tau)| \geq \frac{T_\sigma}{T_c^2}$. According to (3), we can check that $U'(\tau)$ is bounded at $\tau = 0$ and $\tau = 1$, which implies that $U'(\tau)$ is bounded for $\tau \in [0,1]$, leading to

$$|U(\tau(n)) - U(\tau(\hat{n}))| = |U'(\tau_0)| \cdot |\tau(n) - \tau(\hat{n})| \geq \frac{T_\sigma}{T_c^2}|\tau(n) - \tau(\hat{n})|, \tag{18}$$

implying $C_2 = \frac{T_\sigma}{T_c^2}$. Using the explicit C_1 and C_2, we finally have

$$|U(\tau(n)) - U(\tau(\hat{n}))| \geq \frac{T_\sigma}{T_c^2}|\tau(n) - \tau(\hat{n})| \geq \Theta\left(\frac{T_\sigma}{\bar{N}K^2\bar{W}^3T_c^2}\right)|N - \hat{N}|.$$

B Proof of Proposition 5.2

Recall that there is a probability of $\Theta(\frac{1}{K})$ that each token x_i is a noisy version of feature $x_k \in \mathcal{X}$. Let $P_{1:N}^{(t)}$ denote the collection of input tokens for $P^{(t)}$, i.e., $\{x_i\}_{i=1}^N$. We derive the following lemma to characterize the size of the token set $\mathcal{X}_k^s$ for $P_{1:N}^{(t)}$.

Lemma B.1. *Suppose $K^3 = O(N)$. For some constant $c \geq \sqrt{\frac{20K^3}{N}}$, define*

$$\mathcal{E}^* := \left\{ P_{1:N}^{(t)} : |\mathcal{X}_k^s| \in \left[p_k N - \frac{cN}{K}, p_k N + \frac{cN}{K} \right] \text{ for } k \in [K] \right\}. \tag{19}$$

Then, we have

$$\mathbb{P}(P_{1:N}^{(t)} \in \mathcal{E}^*) \geq 1 - 3\exp\left(-\frac{c^2 N}{25K^2}\right).$$

Proof. Note that $|\mathcal{X}_k| \sim \text{multinomial}(N, p_1, \cdots, p_K)$. Let $\delta = \frac{c}{K}$, such that $\frac{\delta^2}{20} \geq \frac{K}{N}$.
According to tail bound of the multinomial distribution, we obtain

$$\mathbb{P}\left(\sum_{i=1}^K \left| |\mathcal{X}_k^s| - \mathbb{E}(|\mathcal{X}_k^s|) \right| > c\frac{N}{K} \right) \leq 3\exp(-\frac{c^2 N}{25K^2}).$$

Since $\mathbb{E}[|\mathcal{X}_k|] = p_k N$, we have

$$\mathbb{P}\left(\cap_{i=1}^K \left\{ \left| |\mathcal{X}_k| - \mathbb{E}(|\mathcal{X}_k|) \right| > c\frac{N}{K} \right\} \right) \leq \mathbb{P}\left(\sum_{i=1}^K \left| |\mathcal{X}_k| - \mathbb{E}(|\mathcal{X}_k|) \right| > c\frac{N}{K} \right) \leq 3\exp(-\frac{c^2 N}{25K^2}),$$

which completes the proof of Lemma B.1. □

For ease of exposition, we define $u_k := (p_k - \delta_0)K$ and $U_k := (p_k + \delta_0)K$, which satisfy $u_k = \Theta(1)$ and $U_k = \Theta(1)$, given $p_k = \Theta(\frac{1}{K})$ and $\delta_0 = O(\frac{1}{K})$. Then for any $P_{1:N}^{(t)}$ belonging to $\mathcal{E}^*$, we have $|\mathcal{X}_k| \in [\frac{u_k N}{K}, \frac{UK}{N}] = \Theta(\frac{N}{K})$ with a probability close to 1. We then prove Proposition 5.2 based on Lemma B.1.

Given $k, k' \in [K]$ with $k' \neq k$, for $t \geq 0$ we define the bilinear attention weights as follows:

$$A_k := x_k^\top Q^{(t)} x_k, \quad \alpha_k^{(t)} = -x_k^\top \nabla_{Q^{(t)}} \mathcal{L}(Q^{(t)}) x_k,$$
$$B_{k,k'} := x_n^\top Q^{(t)} x_k, \quad \beta_{k,k'}^{(t)} = -x_n^\top \nabla_{Q^{(t)}} \mathcal{L}(Q^{(t)}) x_k.$$

Note that the bilinear attention weights characterize the attention scores between any two tokens in eq. (7). Then our Algorithm 1 achieves convergence if and only if the bilinear attention weights converge, i.e., the gradients $\alpha_k^{(t)} = O(\delta_0)$ and $\beta_{k,k'}^{(t)} = O(\delta_0)$. Consequently, we next prove that the two conditions are equivalent with $1 - \text{Attn}_k^{(t)} = O(\delta_0)$.

In the following, we first derive the expressions of $\alpha_k^{(t)}$ and $\beta_{k,k'}^{(t)}$, respectively. By gradient descent (GD) update, we have

$$A_k^{(t+1)} := A_k^{(t)} + \eta \alpha_k^{(t)}$$
$$B_{k,k'}^{(t+1)} := B_{k,k'}^{(t)} + \eta \beta_{k,k'}^{(t)}.$$

Next, we try to derive $\alpha_k^{(t)}$ and $\beta_{k,k'}^{(t)}$. We calculate

$$x_{k'}^\top \nabla_{Q^{(t)}} \mathcal{L} x_k = \mathbb{E}\Big[\mathbb{1}\{x_q^s = x_k\} (\hat{W}_q^s - f^s(x_k)) \sum_{i,j \in [N]} \text{attn}_i^{(t)} \text{attn}_j^{(t)} f^s(x_i) x_{k'}^\top (E_i^x - E_j^x) \Big],$$

which is because ${E_{N+1}^x}^\top x_k \neq 0$ if and only if $x_q^s = x_k$. Then we continue calculating

$$\begin{aligned}
x_{k'}^\top \nabla_{Q^{(t)}} \mathcal{L} x_k =& \mathbb{E}\Big[\mathbb{1}\{x_q^s = x_k\}(\hat{W}_q^s - f^s(x_k)) \sum_{m,n\in[K]} \sum_{i\in\mathcal{X}_m} \sum_{j\in\mathcal{X}_n} \mathrm{attn}_i^{(t)} \mathrm{attn}_j^{(t)} f^s(x_i) x_{k'}^\top (x_m - x_n)\Big] \\
=& \mathbb{E}\Big[\mathbb{1}\{x_q^s = x_k\}(\hat{W}_q^s - f^s(x_k)) \sum_{n\in[K]} \sum_{i\in\mathcal{X}_{k'}} \sum_{j\in\mathcal{X}_n} \mathrm{attn}_i^{(t)} \mathrm{attn}_j^{(t)} f^s(x_i) x_{k'}^\top (x_{k'} - x_n)\Big] \\
&+ \mathbb{E}\Big[\mathbb{1}\{x_q^s = x_k\}(\hat{W}_q^s - f^s(x_k)) \sum_{m\in[K]} \sum_{i\in\mathcal{X}_m} \sum_{j\in\mathcal{X}_{k'}} \mathrm{attn}_i^{(t)} \mathrm{attn}_j^{(t)} f^s(x_i) x_{k'}^\top (x_m - x_{k'})\Big] \\
=& \mathbb{E}\Big[\mathbb{1}\{x_q^s = x_k\}(\hat{W}_q^s - f^s(x_k)) \mathrm{Attn}_{k'}^{(t)} f^s(x_{k'}) \sum_{n\in[K]} \mathrm{Attn}_n^{(t)}\Big] \\
&- \mathbb{E}\Big[\mathbb{1}\{x_q^s = x_k\}(\hat{W}_q^s - f^s(x_k)) \mathrm{Attn}_{k'}^{(t)} \sum_{m\in[K]} \mathrm{Attn}_m^{(t)} f^s(x_m)\Big] \\
=& \mathbb{E}\Big[\mathbb{1}\{x_q^s = x_k\}(\hat{W}_q^s - f^s(x_k)) \mathrm{Attn}_{k'}^{(t)} \sum_{m\in[K]} \mathrm{Attn}_m^{(t)} (f^s(x_{k'}) - f^s(x_m))\Big].
\end{aligned}$$

Since $\hat{W}_q^s = \sum_{i\in[N]} \mathrm{attn}_i f^s(x_i) = \sum_{m\in[K]} \mathrm{Attn}_m^{(t)} f^s(x_m)$, we obtain

$$\begin{aligned}
x_{k'}^\top \nabla_{Q^{(t)}} \mathcal{L} x_k &= -\mathbb{E}\Big[\mathbb{1}\{x_q^s = x_k\} \mathrm{Attn}_{k'}^{(t)} \sum_{n\in[K]} \sum_{m\in[K]} \mathrm{Attn}_m^{(t)} \mathrm{Attn}_n^{(t)} (f^s(x_k) - f^s(x_n))(f^s(x_{k'}) - f^s(x_m))\Big] \\
&= -\mathbb{E}\Big[\mathbb{1}\{x_q^s = x_k\} \mathrm{Attn}_{k'}^{(t)} (f^s(x_k) - \sum_{n\in[K]} \mathrm{Attn}_n^{(t)} f^s(x_n))(f^s(x_{k'}) - \sum_{m\in[K]} \mathrm{Attn}_m^{(t)} f^s(x_m))\Big].
\end{aligned}$$

If $k' = k$, we obtain

$$\begin{aligned}
\alpha_k^{(t)} &= -x_k^\top \nabla_{Q^{(t)}} \mathcal{L} x_k \\
&= \mathbb{E}\Big[\mathbb{1}\{x_q^s = x_k\} \mathrm{Attn}_k^{(t)} \Big(f^s(x_k) - \sum_{n\in[K]} \mathrm{Attn}_n^{(t)} f^s(x_n)\Big)^2\Big].
\end{aligned} \tag{20}$$

As $\alpha_k^{(t)} \geq 0$ is always true, $A_k^{(t)}$ increases with t.

If $k' \neq k$, we obtain

$$\begin{aligned}
\beta_{k,k'}^{(t)} &= -x_{k'}^\top \nabla_{Q^{(t)}} \mathcal{L} x_k \\
&= \mathbb{E}\Big[\mathbb{1}\{x_q^s = x_k\} \mathrm{Attn}_{k'}^{(t)} \Big(f^s(x_k) f^s(x_{k'}) - (f^s(x_k) + f^s(x_k')) \sum_{m\in[K]} \mathrm{Attn}_m^{(t)} f^s(x_m) + \big(\sum_{m\in[K]} \mathrm{Attn}_m^{(t)} f^s(x_m)\big)^2\Big)\Big] \\
&= \mathbb{E}\Big[\mathbb{1}\{x_q^s = x_k\} \mathrm{Attn}_{k'}^{(t)} (f^s(x_k) - \sum_{n\in[K]} \mathrm{Attn}_n^{(t)} f^s(x_n))(f^s(x_{k'}) - \sum_{m\in[K]} \mathrm{Attn}_m^{(t)} f^s(x_m))\Big].
\end{aligned} \tag{21}$$

Based on the expressions of $\alpha_k^{(t)}$ and $\beta_{k,k'}^{(t)}$ above, we next prove the system convergences if and only if $1 - \mathrm{Attn}_k^{(t)} = O(\delta_0)$. We first prove that if $1 - \mathrm{Attn}_k^{(t)} = O(\delta_0)$, the system achieves convergence. Then we prove that if the system achieves convergence, the attention score satisfies $1 - \mathrm{Attn}_k^{(t)} = O(\delta_0)$ for any $k \in [K]$. Then based on the two conclusions, we can derive Proposition 5.2.

We suppose $x_q^s = x_k$ in the following. If $|\mathrm{Attn}_k^{(t)} - 1| = O(\delta_0)$, we have $\sum_{n\in[K], n\neq k} \mathrm{Attn}_n^{(t)} = 1 - \mathrm{Attn}_k^{(t)} = O(\delta_0)$, which means $\mathrm{Attn}_{k'} = O(\delta_0)$ for any $k' \neq k$. Then based on (20) and (21), we calculate

$$\begin{aligned}
\alpha_k^{(t)} =& \mathbb{E}\Big[\mathrm{Attn}_k^{(t)} \Big(f^s(x_k) - \sum_{n\in[K]} \mathrm{Attn}_n^{(t)} f^s(x_n)\Big)^2\Big] \\
=& \mathbb{E}\Big[\Theta(1) \cdot \Big(\sum_{n\in[K]} \big(\mathrm{Attn}_n^{(t)} f^s(x_k) - \mathrm{Attn}_n^{(t)} f^s(x_n)\big)\Big)^2\Big] \\
=& O(\delta_0).
\end{aligned}$$

Similarly, we calculate

$$\begin{aligned}\beta_{k,k'}^{(t)} =&\mathbb{E}\Big[\mathrm{Attn}_{k'}^{(t)}(f^s(x_k)-\sum_{n\in[K]}\mathrm{Attn}_n^{(t)}f^s(x_n))(f^s(x_{k'})-\sum_{m\in[K]}\mathrm{Attn}_m^{(t)}f^s(x_m))\Big]\\ =&\mathbb{E}\Big[\mathrm{Attn}_{k'}^{(t)}(f^s(x_k)-\mathrm{Attn}_k f^s(x_k)+O(\delta_0))(f^s(x_{k'})-\sum_{m\in[K]}\mathrm{Attn}_m^{(t)}f^s(x_m))\Big]\\ =&O(\delta_0).\end{aligned}$$

Consequently, the system has achieved the convergence state.

If the system has achieved the convergence state at time T_s, we prove (11) by contradiction. At time $t > T_s$, we assume that the attention score $\mathrm{Attn}_k^{(t)}$ of the query token $x_q^s = x_k$ satisfies $1 - \mathrm{Attn}_k^{(t)} = \Theta(1)$. Then there must exist another $k' \in [K]$ with $k' \neq k$ so that $\mathrm{Attn}_{k'}^{(t)} = \Theta(1)$ for the current prompt. Then we calculate

$$\begin{aligned}\alpha_k^{(t)} =&\mathbb{E}\Big[\mathrm{Attn}_k^{(t)}\Big(f^s(x_k)-\sum_{n\in[K]}\mathrm{Attn}_n^{(t)}f^s(x_n)\Big)^2\Big]\\ =&\mathbb{E}\Big[\Theta(1)\cdot\Big(\sum_{n\in[K]}\big(\mathrm{Attn}_n^{(t)}f^s(x_k)-\mathrm{Attn}_n^{(t)}f^s(x_n)\big)\Big)^2\Big]\\ \geq&\mathbb{E}\Big[\Theta(1)\cdot\big(\mathrm{Attn}_{k'}^{(t)}(f^s(x_k)-f^s(x_{k'}))\big)^2\Big]\\ =&\Omega(\delta_0),\end{aligned}$$

where the second equality is because of $\mathrm{Attn}_k^{(t)} = \Theta(1)$, the last equality is because of $\|x_k - x_{k'}\| = \Theta(\Delta)$ and Assumption 1. Given the gradient $\alpha_k^{(t)} = \Omega(\delta_0)$, we have $|\mathrm{Attn}_k^{(t+1)} - \mathrm{Attn}_k^{(t)}| = \Omega(\Delta^2)$, which is contradicted with the convergence state $\alpha_k^{(t)} = O(\delta_0)$. Consequently, if the system has achieved the convergence state, (11) always holds, which completes the proof of Proposition 5.2.

C Proof of Theorem 5.3

In this proof, we suppose $x_q^s = x_k$. We first prove that the attention score $\mathrm{Attn}_k^{(t)}$ in (10) increases to $\mathrm{Attn}_k^{(T^*)} = \Omega(\frac{1}{1+\delta_0\epsilon})$ at time $T^* = \Theta(\frac{K\log(K\epsilon^{-1})}{\eta\delta_0^2L^2\Delta^2})$. Then we prove at time T^*, this achieved attention score ensures that the prediction loss satisfies $\mathcal{L}(\theta^{T^*}) = O(\epsilon^2)$.

C.1 Growth of the target attention score $\mathrm{Attn}_k^{(t)}$

We study the growth of $\mathrm{Attn}_k^{(t)}$ in two learning stages:

- In the first stage, where $t \in \{1, \cdots, T_1\}$ with $T_1 = \Theta(\frac{K\log(K)}{\eta L^2\Delta^2})$, the bilinear attention weight $A_k^{(t)} = x_k^\top Q^{(t)} x_k$ increases at a rate of $\Theta(\frac{\eta L^2\Delta^2}{K})$. After the end of the first stage, we have $\mathrm{Attn}_k^{(T_1+1)} = \Omega(\frac{1}{1+\delta_0})$.
- In the second stage, where $t \in \{T_1+1, \cdots, T^*\}$ with $T^* = \Theta(\frac{K\log(K\epsilon^{-1})}{\eta\delta_0^2L^2\Delta^2})$, $A_k^{(t)}$ increases at a rate of $\Theta(\frac{\eta\delta_0^2L^2\Delta^2}{K})$, and we obtain $\mathrm{Attn}_k^{(T^*)} = \Omega(\frac{1}{1+\delta_0\epsilon})$ at the end of the second stage.

We first rewrite the gradient $\alpha_k^{(t)}$ as follows:

$$\begin{aligned}\alpha_k^{(t)} =&\mathbb{E}\Big[\mathbb{1}\{x_q^s = x_k\}\mathrm{Attn}_k^{(t)}\Big(f^s(x_k)-\sum_{n\in[K]}\mathrm{Attn}_n^{(t)}f^s(x_n)\Big)^2\Big]\\ =&\mathbb{E}\Big[\mathbb{1}\{x_q^s = x_k\}\mathrm{Attn}_k^{(t)}\Big(\sum_{n\in[K]}\mathrm{Attn}_n^{(t)}\big(f^s(x_k)-f^s(x_n)\big)\Big)^2\Big]\\ \geq&\mathbb{E}\Big[\mathbb{1}\{x_q^s = x_k\}\mathrm{Attn}_k^{(t)}\Big(\sum_{n\in[K]}\mathrm{Attn}_n^{(t)}\min_{k,n\in[K]}|f^s(x_k)-f^s(x_n)|\Big)^2\Big]\\ =&\mathbb{E}\Big[\mathbb{1}\{x_q^s = x_k\}\mathrm{Attn}_k^{(t)}\Big(1-\mathrm{Attn}_k^{(t)}\Big)^2\Theta(L^2\Delta^2)\Big],\end{aligned}$$

where the last equality is due to Assumption 1. Consequently, we obtain $\alpha_k^{(t)} = \mathbb{E}\Big[\mathbb{1}\{x_q^s = x_k\}\mathrm{Attn}_k^{(t)}\Big(1-\mathrm{Attn}_k^{(t)}\Big)^2\Theta(L^2\Delta^2)\Big]$.

For the first stage with $t \in \{1, \cdots, T_1\}$, based on the gradient expression above, we calculate

$$\begin{aligned}\alpha_k^{(t)} =&\mathbb{E}\Big[\mathbb{1}\{x_q^s = x_k\}\mathrm{Attn}_k^{(t)}\Big(1-\mathrm{Attn}_k^{(t)}\Big)^2\Theta(L^2\Delta^2)\Big]\\ =&p_k\cdot\mathbb{E}\left[\mathrm{Attn}_k^{(t)}\Big(1-\mathrm{Attn}_k^{(t)}\Big)^2\Big|x_q^s=x_k\}\right]\cdot\Theta(L^2\Delta^2)\\ =&\Theta(\frac{L^2\Delta^2}{K}).\end{aligned}$$

where the last equality is because of $p_k = \Theta(\frac{1}{K})$, $\mathrm{Attn}_k^{(t)} = \Theta(1)$ and $1-\mathrm{Attn}_k^{(t)} = \Theta(1)$ for $t \in \{1, \cdots, T_1\}$. For any $k \neq n$, we calculate

$$\begin{aligned}|\beta_{k,n}^{(t)}| &= \mathbb{E}\Big[\mathbb{1}\{x_q^s = x_k\}\mathrm{Attn}_n^{(t)}|f^s(x_k) - \sum_{j\in[K]}\mathrm{Attn}_j^{(t)}f^s(x_j)|\cdot|f^s(x_n) - \sum_{m\in[K]}\mathrm{Attn}_m^{(t)}f^s(x_m)|\Big]\\ &= p_k\cdot\mathbb{E}\Big[\mathrm{Attn}_n^{(t)}\Big|\sum_{j\in[K]}(\mathrm{Attn}_j^{(t)}f^s(x_k) - \mathrm{Attn}_j^{(t)}f^s(x_j))\Big|\cdot\Big|\sum_{m\in[K]}(\mathrm{Attn}_m^{(t)}f^s(x_n) - \mathrm{Attn}_m^{(t)}f^s(x_m))\Big|\Big]\\ &< p_k\cdot\mathbb{E}\Big[\mathrm{Attn}_n^{(t)}\cdot\sum_{j\in[K]}\mathrm{Attn}_j^{(t)}|f^s(x_k) - f^s(x_j)|\cdot\sum_{m\in[K]}\mathrm{Attn}_m^{(t)}|f^s(x_n) - f^s(x_m))|\Big]\\ &= p_k\cdot\mathbb{E}\left[\mathrm{Attn}_n^{(t)}\cdot(1-\mathrm{Attn}_k^{(t)})\cdot(1-\mathrm{Attn}_n^{(t)})\cdot\Theta(L^2\Delta^2)\right]\\ &= O(\frac{L^2\Delta^2}{K^2}),\end{aligned}$$

where the inequality is derived by union bound, and the last equality is because of $|f^s(x_k) - f^s(x_j)| = |f^s(x_n) - f^s(x_m)| = \Theta(L\Delta)$ for any $j \neq k$ and $m \neq n$, respectively, and the last equality is because of $\mathrm{Attn}_n^{(t)} = \Theta(\frac{1}{K})$, $p_k = \Theta(\frac{1}{K})$, and $1-\mathrm{Attn}_k^{(t)} < 1$.

Then we calculate

$$\begin{aligned}A_k^{(T_1+1)} &= A_k^{(T_1)} + \eta\alpha_k^{(T_1)} = A_k^{(T_1-1)} + \eta\alpha_k^{(T_1-1)} + \eta\alpha_k^{(T_1)}\\ &= \cdots = A_k^{(0)} + \eta\cdot\Theta(\frac{L^2\Delta^2}{K})\cdot T_1 = \Theta(\log(K)).\end{aligned}$$

Given $|\beta_{k,m}^{(t)}| = O(\frac{L^2C^2}{K^2})$, we can similarly calculate

$$\begin{aligned}B_{k,m}^{(T_1+1)} &= B_{k,m}^{(T_1)} + \eta\beta_{k,m}^{(T_1)}\\ &\le |B_{k,m}^{(T_1)}| + \eta|\beta_{k,m}^{(T_1)}|\\ &\le |B_{k,m}^{(0)}| + \eta\cdot O(\frac{L^2\Delta^2}{K^2})\cdot T_1\\ &= O(\frac{\log(K)}{K}).\end{aligned}$$

Consequently, we have $B_{k,m}^{(T_1+1)} = O(\frac{\log(K)}{K})$. Finally, we calculate the attention score

$$\begin{aligned}\mathrm{Attn}_k^{(t)} &= \frac{|\mathcal{X}_k|e^{x_k^\top Q^{(t)}x_k}}{\sum_{j\in[N]}e^{E_j^{x\top}Q^{(t)}x_k}}\\ &= \frac{|\mathcal{X}_k|e^{x_k^\top Q^{(t)}x_k}}{\sum_{m\neq k}|\mathcal{X}_m|e^{x_m{}^\top Q^{(t)}x_k} + |\mathcal{X}_k|e^{x_k^\top Q^{(t)}x_k}}\\ &= \frac{1}{\sum_{m\neq k}\frac{|\mathcal{X}_m|}{|\mathcal{X}_k|}\exp\left(B_{k,m}^{(t)} - A_k^{(t)}\right) + 1}.\\ &\ge \frac{1}{O(\frac{1}{K})(\frac{N}{|\mathcal{X}_k|}-1)+1}\\ &\ge \frac{1}{O(\frac{1}{u_k}-\frac{1}{K})+1}\\ &= \Omega(\frac{1}{1+\delta_0}),\end{aligned}$$

where the first inequality is because of $\exp\left(B_{k,m}^{(t)} - A_k^{(t)}\right) \leq \exp\left(\frac{\log(K)}{K} - \log(K)\right) \leq O(\frac{1}{K})$, and the last equality is because of $\frac{1}{u_k} - \frac{1}{K} = \Theta(\delta_0)$ derived in Lemma B.1.

For the second stage with $t \in \{T_1 + 1, \cdots, T^*\}$, we can use the similar way to calculate

$$\alpha_k^{(t)} = \Theta(\frac{\delta^2 L^2 \Delta^2}{K}), \quad |\beta_{k,m}^{(t)}| = O(\frac{\delta^2 L^2 \Delta^2}{K^2}).$$

Then at the end of the second stage, we obtain

$$A_k^{(T^*)} = \Theta(\log(K\epsilon^{-1})), \quad B_{k,m}^{(T^*)} = \Theta(\frac{\log(K\epsilon^{-1})}{K}).$$

We further calculate

$$\mathrm{Attn}_k^{(T^*)} \geq \frac{1}{O(\frac{\epsilon}{K})(\frac{N}{|\mathcal{X}_k|} - 1) + 1} \geq \frac{1}{O(\epsilon) \cdot O(\frac{1}{u_k} - \frac{1}{K}) + 1} = \Omega(\frac{1}{1 + \epsilon\delta_0}),$$

C.2 Convergence of prediction loss $\mathcal{L}(\theta)$

We rewrite the prediction error function $\mathcal{L}^{(t)}(\theta)$ in (9) into:

$$\begin{aligned}
\mathcal{L}^{(t)}(\theta) =& \frac{1}{2}\sum_{k=1}^{K} \mathbb{E}\left[\mathbb{1}\{x_q^s = x_k\}(\hat{W}_q^s - f^s(x_k))^2\right] \\
=& \frac{1}{2}\sum_{k=1}^{K} \mathbb{E}\left[\mathbb{1}\{x_q^s = x_k\}\Big(\sum_{n\in[K]} \mathrm{Attn}_n^{(t)} f^s(x_n) - f^s(x_k)\Big)^2\right] \\
=& \frac{1}{2}\sum_{k=1}^{K} \mathbb{E}\left[\mathbb{1}\{x_q^s = x_k\}\Big(\sum_{n\neq k} \mathrm{Attn}_n^{(t)} \left(f^s(x_n) - f^s(x_k)\right)\Big)^2\right] \\
=& \frac{1}{2}\sum_{k=1}^{K} \mathbb{E}\left[\mathbb{1}\{x_q^s = x_k\}\Big(1 - \mathrm{Attn}_k^{(t)}\Big)^2 \Theta(L^2\Delta^2)\right],
\end{aligned}$$

where the last equality is because of $\sum_{n\neq k} \mathrm{Attn}_n^{(t)} = 1 - \mathrm{Attn}_k^{(t)}$ and $\left|f^s(x_n) - f^s(x_k)\right| = \Theta(L\Delta)$.

Suppose $x_q^s = x_k$ at time T^*. Based on the conclusion $1 - \mathrm{Attn}_k^{(T^*)} = O(\epsilon\delta_0)$ derived in Appendix C.1, we finally calculate

$$\begin{aligned}
\mathcal{L}^{(T^*)}(\theta) &= \frac{1}{2}\sum_{k=1}^{K} \mathbb{E}\left[\mathbb{1}\{x_q^s = x_k\}\Big(1 - \mathrm{Attn}_k^{(T^*)}\Big)^2 \Theta(L^2\Delta^2)\right] \\
&= \mathbb{E}\left[\Big(1 - \mathrm{Attn}_k^{(T^*)}\Big)^2 \Theta(L^2\Delta^2)\right] \\
&= O(\epsilon^2),
\end{aligned}$$

where the last equality is because of $\Theta(L^2\Delta^2) \cdot O(\epsilon^2\delta_0^2) = O(\epsilon^2)$ given $L \leq \Theta(\frac{1}{L\Delta})$. This completes the proof of Theorem 5.3.

D Proof of Lemma 5.4

To prove the upper bound in Lemma 5.4, we first prove that there exists a $C_3 > 0$ such that $|\tau(W_k) - \tau(\hat{W}_k)| \leq C_3|W_k - \hat{W}_k|$. Then, we prove that there exists a $C_4 > 0$ such that $|U(\tau(W_k)) - U(\tau(\hat{W}_k))| \leq C_4|\tau(W_k) - \tau(\hat{W}_k)|$.

To find the exact C_3, we define:

$$f^s(\tau, \boldsymbol{W}) := \tau(1-\tau)^{N-1} \sum_{k=0}^{K-1} (1 - (1-\tau)^{N-1})^k W_k + \tau(1 - (1-\tau)^{N-1})^K W_K + \tau - 2,$$

where $\boldsymbol{W} = \{W_k\}_{k=0}^K$ is the vector of contention window thresholds. We can check that

$$\left|\frac{\partial f^s(\tau, \boldsymbol{W})}{\partial \tau}\right| \geq \left|\frac{\partial f^s(\tau, \boldsymbol{W})}{\partial \tau}\Big|_{W_k=1}\right| \geq 2.$$

The derivative of F on W_k is

$$\frac{\partial f^s(\tau, \boldsymbol{W})}{\partial W_k} = \begin{cases} \tau(1-\tau)^{N-1}(1 - (1-\tau)^{N-1})^k, & \text{if } k = 0, \cdots, K-1, \\ \tau(1 - (1-\tau)^{N-1})^K, & \text{if } k = K. \end{cases}$$

Note that

$$\tau(1-\tau)^{N-1}(1-(1-\tau)^{N-1})^k \le \tau(1-\tau)^{N-1} \le \frac{1}{N}\left(1-\frac{1}{N}\right)^{N-1} \le \frac{1}{4}, \tag{22}$$

where the first inequality holds due to $(1-(1-\tau)^{N-1})^k \le 1$, the second due to $\tau(1-\tau)^{N-1}$ increasing with $\tau \le \frac{1}{N}$ (this bound $1/N$ comes from Lemma A.1), and the last due to $\frac{1}{N}(1-\frac{1}{N})^{N-1}$ decreasing with $N \ge 2$. Further, we have

$$\tau(1-(1-\tau)^{N-1})^K \le \frac{1}{N}\left(1-\left(1-\frac{1}{N}\right)^{N-1}\right)^K \le \frac{1}{2}\left(1-\left(1-\frac{1}{\bar{N}}\right)^{\bar{N}-1}\right)^K \le \frac{1}{2}\left(1-\frac{1}{e}\right)^K < \frac{1}{3}, \tag{23}$$

where the first inequality holds due to $\tau(1-(1-\tau)^{N-1})^K$ increasing with $\tau \le \frac{1}{N}$, the second due to $N \in [2, \bar{N}]$ and $(1-(1-\frac{1}{N})^{N-1})^K$ increasing with N, the third due to $(1-\frac{1}{\bar{N}})^{\bar{N}-1} \ge 1/e$, the last due to $(1-\frac{1}{e})^K < \frac{2}{3}$. According to (22) and (23), we can now upper bound $\frac{\partial f^s(\tau,\boldsymbol{W})}{\partial W_k}$ as follows:

$$\frac{\partial f^s(\tau,\boldsymbol{W})}{\partial W_k} \le \frac{1}{4}.$$

Based on the Implicit Function Theorem, we have

$$\left|\frac{\partial \tau}{\partial W_k}\right| = \frac{|\frac{\partial f^s(\tau,\boldsymbol{W})}{\partial W_k}|}{|\frac{\partial f^s(\tau,\boldsymbol{W})}{\partial \tau}|} \le \frac{1}{8},$$

implying

$$|\tau(W_k) - \tau(\hat{W}_k)| \le \frac{1}{8}|W_k - \hat{W}_k|$$

according to the Mean Value Theorem. Then we can set $C_3 = \frac{1}{8}$. Following a similar analysis, we can upper bound $|U'(\tau)|$ as

$$|U'(\tau)| \le \frac{T_P\bar{N}}{T_\sigma} = C_4,$$

implying

$$|U(\tau(W_k)) - U(\tau(\hat{W}_k))| \le \frac{T_P\bar{N}}{T_\sigma}|\tau(W_k) - \tau(\hat{W}_k)| \le \frac{T_P\bar{N}}{8T_\sigma}|W_k - \hat{W}_k|.$$

E Proof of Theorem 5.5

According to Theorem 5.3, we have

$$\mathcal{L}(\theta^{(T^*)}) = \mathbb{E}\left[\left(\hat{W}_q^s - W_q^s\right)^2\right] \le \mathcal{O}(\epsilon^2). \tag{24}$$

Thus, we have

$$\begin{aligned}
\Delta U &= \mathbb{E}\left[\left(U(W_q^s) - U(\hat{W}_q^s)\right)\right] \\
&= \mathbb{E}|U(W_q^s) - U(\hat{W}_q^s)| \\
&\le \mathbb{E}\left[\frac{T_P\bar{N}}{8T_\sigma} \cdot |W_q^s - \hat{W}_q^s|\right] \\
&= \frac{T_P\bar{N}}{8T_\sigma}\mathbb{E}\left|W_q^s - \hat{W}_q^s\right| \\
&\le \frac{T_P\bar{N}}{8T_\sigma}\left(\mathbb{E}\left(W_q^s - \hat{W}_q^s\right)^2\right)^{\frac{1}{2}} \\
&\le \frac{T_P\bar{N}}{8T_\sigma}\left(\mathbb{E}\left(W_q^s - \hat{W}_q^s\right)^2\right)^{\frac{1}{2}} \\
&\le \frac{T_P\bar{N}}{8T_\sigma} \cdot \mathcal{O}(\epsilon) = \mathcal{O}\left(\frac{T_P\bar{N}\epsilon}{8T_\sigma}\right),
\end{aligned}$$

where the first inequality holds due to Lemma 5.4, the second and the third hold due to Jensen's inequality, and the last due to (24). We then finish the proof.

F Proof of Lemma 6.1

Before the formal proof, let us introduce a useful lemma in the following.

Lemma F.1. *Suppose that the minimum KL-divergence between our ICL mapping f and the ground-truth f^* satisfies $\min_f KL(\mathbb{P}_f, \mathbb{P}_{f^*}) > -8\ln(\alpha\beta)$. If the number of in-context data examples M is long enough as*

$$M \geq \max\left\{\frac{-(\ln q)(16\ell^2)(\ln^2\beta)}{KL^2(\mathbb{P}_f, \mathbb{P}_{f^*})}, \frac{-2\ln\mu}{\min_f KL(\mathbb{P}_f, \mathbb{P}_{f^*}) + 8\ln(\alpha\beta)}\right\}$$

for any $\mu > 0$, $q \in (0,1)$ and any mapping $f \neq f^*$, we have

$$Pr\left(\frac{\mathbb{P}_f(P)}{\mathbb{P}_{f^*}(P)} < \mu\right) \geq 1 - q.$$

In Section F.1, we first prove Lemma F.1. Then, we prove Lemma 6.1 in Section F.2.

F.1 Proof of Lemma F.1

Given

$$\alpha\mathbb{P}_f(s_2|s_1 \oplus \text{“\textbackslash n”}) \leq \mathbb{P}_f(s_2) \leq \frac{1}{\alpha}\mathbb{P}_f(s_2|s_1 \oplus \text{“\textbackslash n”}),\ \alpha \in (0,1].$$

with $s_1 = x_1 \oplus W_1 \oplus \cdots \oplus x_m \oplus W_m$ and $s_2 = x_{m+1} \oplus W_{m+1}$, $m \leq M-1$, we have

$$\alpha \leq \frac{\mathbb{P}_f(s_1 \oplus \text{“\textbackslash n”}) \cdot \mathbb{P}_f(s_2)}{\mathbb{P}_f(s_1 \oplus \text{“\textbackslash n”} \oplus s_2)} \leq \frac{1}{\alpha}.$$

By multiplying $\frac{\mathbb{P}_f(s_1 \oplus \text{“\textbackslash n”}) \cdot \mathbb{P}_f(s_2)}{\mathbb{P}_f(s_1 \oplus \text{“\textbackslash n”} \oplus s_2)}$ for all the possible s_1 and s_2, we obtain the following inequality:

$$\alpha^M \leq \frac{\prod_{m=1}^M \mathbb{P}_f(x_m \oplus W_m \oplus \text{“\textbackslash n”})}{\mathbb{P}_f(x_1 \oplus W_1 \oplus \text{“\textbackslash n”} \oplus \cdots \oplus x_M \oplus W_M \oplus \text{“\textbackslash n”})} \leq \alpha^{-M}. \tag{25}$$

Further, we have

$$\mathbb{P}_f(x_m \oplus W_m) = \mathbb{P}_f(x_m)\mathbb{P}_f(W_m|x_m) > \mathbb{P}_f(x_m)\mathbb{P}_f(W_m^*|x_m)\mathbb{P}_f(W_m|W_m^*) = \mathbb{P}_f(x_m \oplus W_m^*)\mathbb{P}_f(W_m|W_m^*) > \mathbb{P}_f(x_m \oplus W_m^*) \cdot \beta, \tag{26}$$

where the inequality holds due to the assumption in Section 6.1. According to (26), we also have

$$\mathbb{P}_f(x_m \oplus W_m^*) > \mathbb{P}_f(x_m \oplus W_m) \cdot \beta. \tag{27}$$

Based on (26) and (27), we have

$$\beta < \frac{\mathbb{P}_f(x_m \oplus W_m^*)}{\mathbb{P}_f(x_m \oplus W_m)} < \beta^{-1}. \tag{28}$$

Since

$$\mathbb{P}_f(x_m \oplus W_m) \geq \mathbb{P}_f(x_m \oplus W_m \oplus \text{“\textbackslash n”}) = \mathbb{P}_f(x_m \oplus W_m) \cdot \mathbb{P}_f(\text{“\textbackslash n”}|x_m \oplus W_m) > \mathbb{P}_f(x_m \oplus W_m) \cdot \beta, \tag{29}$$

where the inequality holds due to the assumption in Section 6.1. Further,

$$\mathbb{P}_f(x_m \oplus W_m \oplus \text{“\textbackslash n”}) < \mathbb{P}_f(x_m \oplus W_m) < \mathbb{P}_f(x_m \oplus W_m) \cdot \beta^{-1} \tag{30}$$

due to $\beta \in (0,1)$. According to (29) and (30), we have

$$\beta < \frac{\mathbb{P}_f(x_m \oplus W_m)}{\mathbb{P}_f(x_m \oplus W_m \oplus \text{“\textbackslash n”})} < \beta^{-1}. \tag{31}$$

According to (25), (28) and (31), we can now obtain that

$$\frac{\prod_{m=1}^M \mathbb{P}_f(x_m \oplus W_m^*)}{\mathbb{P}_f(x_1 \oplus W_1 \oplus \text{“\textbackslash n”} \oplus \cdots \oplus x_M \oplus W_M \oplus \text{“\textbackslash n”})} = \frac{\prod_{m=1}^M \mathbb{P}_f(x_m \oplus W_m \oplus \text{“\textbackslash n”})}{\mathbb{P}_f(x_1 \oplus W_1 \oplus \text{“\textbackslash n”} \oplus \cdots \oplus x_M \oplus W_M \oplus \text{“\textbackslash n”})} \cdot \frac{\prod_{m=1}^M \mathbb{P}_f(x_m \oplus W_m)}{\prod_{m=1}^M \mathbb{P}_f(x_m \oplus W_m \oplus \text{“\textbackslash n”})} \cdot \frac{\prod_{m=1}^M \mathbb{P}_f(x_m \oplus W_m^*)}{\prod_{m=1}^M \mathbb{P}_f(x_m \oplus W_m)}$$
$$\in [\alpha^M\beta^{2M}, \alpha^{-M}\beta^{-2M}]. \tag{32}$$

Denote $P = x_1 \oplus W_1 \oplus \text{“\textbackslash n”} \oplus \cdots \oplus x_M \oplus W_M \oplus \text{“\textbackslash n”}$. We have

$$\begin{aligned}\ln \frac{\mathbb{P}_f(P)}{\mathbb{P}_{f^*}(P)} &= \ln \frac{\mathbb{P}_f(P)}{\prod_{m=1}^M \mathbb{P}_f(x_m \oplus W_m^*)} + \ln \frac{\prod_{m=1}^M \mathbb{P}_f(x_m \oplus W_m^*)}{\prod_{m=1}^M \mathbb{P}_{f^*}(x_m \oplus W_m^*)} + \ln \frac{\prod_{m=1}^M \mathbb{P}_{f^*}(x_m \oplus W_m^*)}{\mathbb{P}_{f^*}(P)} \\ &\le \ln \alpha^{-M}\beta^{-2M} + \ln \frac{\prod_{m=1}^M \mathbb{P}_f(x_m \oplus W_m^*)}{\prod_{m=1}^M \mathbb{P}_{f^*}(x_m \oplus W_m^*)} + \ln \alpha^{-M}\beta^{-2M} \\ &= 4M \ln \alpha^{-1}\beta^{-1} + \sum_{m=1}^M \ln \frac{\mathbb{P}_f(x_m \oplus W_m^*)}{\mathbb{P}_{f^*}(x_m \oplus W_m^*)},\end{aligned}$$

where the inequality holds due to (32). Note that

$$\mathbb{E}\left[\frac{1}{M}\sum_{m=1}^M \ln \frac{\mathbb{P}_f(x_m \oplus W_m^*)}{\mathbb{P}_{f^*}(x_m \oplus W_m^*)}\right] = -KL(\mathbb{P}_{f^*}, \mathbb{P}_f), \tag{33}$$

and

$$\left|\ln \frac{\mathbb{P}_f(x_m \oplus W_m^*)}{\mathbb{P}_{f^*}(x_m \oplus W_m^*)}\right| = \left|\sum_{l=1}^{\ell} \ln \frac{\mathbb{P}_f(x_m^{l+1}|x_m^{1:l})}{\mathbb{P}_{f^*}(x_m^{l+1}|x_m^{1:l})}\right| \le \sum_{l=1}^{\ell}\left|\ln \frac{\mathbb{P}_f(x_m^{l+1}|x_m^{1:l})}{\mathbb{P}_{f^*}(x_m^{l+1}|x_m^{1:l})}\right| \le \sum_{l=1}^{\ell} \ln \beta^{-1} = \ell \ln \beta^{-1}, \tag{34}$$

where the second inequality holds due to

$$\frac{\mathbb{P}_f(x_m^{l+1}|x_m^{1:l})}{\mathbb{P}_{f^*}(x_m^{l+1}|x_m^{1:l})} \le \frac{1}{\mathbb{P}_{f^*}(x_m^{l+1}|x_m^{1:l})} \le \frac{1}{\beta} \text{ with } \mathbb{P}_{f^*}(x_m^{l+1}|x_m^{1:l}) > \beta.$$

Based on (33) and (34), according to the Hoeffding Inequality, we have

$$Pr\left(\frac{\mathbb{P}_f(P)}{\mathbb{P}_{f^*}(P)} \le e^{-M(KL(\mathbb{P}_{f^*},\mathbb{P}_f)-\mu'-4\ln\alpha^{-1}\beta^{-1})}\right) \ge 1 - e^{\frac{-2M\mu'}{(2\ell\ln\beta)^2}}. \tag{35}$$

We take $\mu' = \frac{1}{2}KL(\mathbb{P}_{f^*}, \mathbb{P}_f)$. Therefore, for any (μ, q) satisfies

$$\mu > e^{\frac{M}{2}(KL(\mathbb{P}_{f^*},\mathbb{P}_f)-8\ln\alpha^{-1}\beta^{-1})},\ q > e^{-\frac{M(KL(\mathbb{P}_{f^*},\mathbb{P}_f))^2}{8\ell^2\ln^2\beta}}, \tag{36}$$

we always have

$$Pr\left(\frac{\mathbb{P}_f(P)}{\mathbb{P}_{f^*}(P)} \le \mu\right) \ge 1 - q.$$

After rewriting (36), we obtain the condition on M as follows:

$$M \ge \max\left\{\frac{-(\ln q)(16\ell^2)(\ln^2\beta)}{KL^2(\mathbb{P}_f, \mathbb{P}_{f^*})}, \frac{-2\ln\mu}{\min_\phi KL(\mathbb{P}_f, \mathbb{P}_{f^*}) + 8\ln(\alpha\beta)}\right\}.$$

F.2 Proof of Lemma 6.1

Denote $P' = x_1 \oplus W_1 \oplus \text{“\textbackslash n”} \oplus \cdots \oplus x_M \oplus W_M \oplus \text{“\textbackslash n”}$ and $P = P' \oplus x$. We have

$$\mathbb{P}_{\mathcal{D}}(W|P' \oplus x) - \mathbb{P}_{\mathcal{D}}(W^*|P' \oplus x) = \frac{\sum_{f\in\mathcal{D}} Pr(f|\mathcal{D})(\mathbb{P}_f(P' \oplus x \oplus W) - \mathbb{P}_f(P' \oplus x \oplus W^*))}{\sum_{f\in\mathcal{D}} Pr(f|\mathcal{D})\mathbb{P}_f(P' \oplus x)}. \tag{37}$$

According to the assumption in Section 6.1, we have

$$\begin{aligned}\alpha \le \frac{\mathbb{P}_f(P' \oplus x \oplus W)}{\mathbb{P}_f(P')\mathbb{P}_f(x \oplus W)} \le \alpha^{-1}, \\ \mathbb{P}_f(P' \oplus x) \le \alpha^{-1}\mathbb{P}_f(P')\mathbb{P}_f(x),\end{aligned} \tag{38}$$

which implies

$$\mathbb{P}_f(P' \oplus x \oplus W) \ge \alpha\mathbb{P}_f(P')\mathbb{P}_f(x \oplus W), \tag{39}$$

$$\mathbb{P}_f(P' \oplus x \oplus W) \le \alpha^{-1}\mathbb{P}_f(P')\mathbb{P}_f(x \oplus W^*). \tag{40}$$

Substitute (38)-(40) into (37), we have

$$\mathbb{P}_{\mathcal{D}}(W|P' \oplus x) - \mathbb{P}_{\mathcal{D}}(W^*|P' \oplus x) \ge \frac{\sum_{f\in\mathcal{D}} Pr(f|\mathcal{D})\mathbb{P}_f(P')(\alpha^2\mathbb{P}_f(x \oplus W) - \alpha^{-2}\mathbb{P}_f(x \oplus W^*))}{\sum_{f\in\mathcal{D}} Pr(f|\mathcal{D})\mathbb{P}_f(P')\mathbb{P}_f(x)}.$$

Define

$$A := Pr(f^*|\mathcal{D})\mathbb{P}_{f^*}(P')(\alpha^2\mathbb{P}_{f^*}(x \oplus W) - \alpha^{-2}\mathbb{P}_{f^*}(x \oplus W^*)),$$
$$B := \sum_{f\in\mathcal{D}, f\neq f^*} Pr(f|\mathcal{D})\mathbb{P}_f(P')(\alpha^2\mathbb{P}_f(x \oplus W) - \alpha^{-2}\mathbb{P}_f(x \oplus W^*)),$$
$$C := Pr(f^*|\mathcal{D})\mathbb{P}_{f^*}(P')\mathbb{P}_{f^*}(x),$$
$$D := \sum_{f\in\mathcal{D}, f\neq f^*} Pr(f|\mathcal{D})\mathbb{P}_f(P')\mathbb{P}_f(x),$$

we then have

$$\mathbb{P}_{\mathcal{D}}(W|P' \oplus x) - \mathbb{P}_{\mathcal{D}}(W^*|P' \oplus x) \geq \frac{A}{C+D} + \frac{B}{C+D}.$$

Next, we derive upper bounds of $|\frac{B}{C}|$ and $|\frac{D}{C}|$. We have

$$\left|\frac{B}{C}\right| = \left|\frac{\sum_{f\in\mathcal{D}, f\neq f^*} Pr(f|\mathcal{D})\mathbb{P}_f(P')(\alpha^2\mathbb{P}_f(x \oplus W) - \alpha^{-2}\mathbb{P}_f(x \oplus W^*))}{Pr(f^*|\mathcal{D})\mathbb{P}_{f^*}(P')\mathbb{P}_{f^*}(x)}\right| \leq \sum_{f\in\mathcal{D}, f\neq f^*}\left|\frac{Pr(f|\mathcal{D})\mathbb{P}_f(P')(\alpha^2\mathbb{P}_f(x \oplus W) - \alpha^{-2}\mathbb{P}_f(x \oplus W^*))}{Pr(f^*|\mathcal{D})\mathbb{P}_{f^*}(P')\mathbb{P}_{f^*}(x)}\right|.$$

Since we have

$$\alpha^2\mathbb{P}_f(x \oplus W) - \alpha^{-2}\mathbb{P}_f(x \oplus W^*) \leq 1 \leq \alpha^{-2},$$

we further bound $|\frac{B}{C}|$ as follows:

$$\left|\frac{B}{C}\right| \leq \sum_{f\in\mathcal{D}, f\neq f^*} \frac{Pr(f|\mathcal{D})}{Pr(f|\mathcal{D}^*)} \cdot \frac{\mathbb{P}_f(P')}{\mathbb{P}_{f^*}(P')} \cdot \alpha^{-2} \cdot \frac{1}{\mathbb{P}_{f^*}(x)} \leq \sum_{f\in\mathcal{D}, f\neq f^*} \frac{1}{\gamma} \cdot \frac{\mathbb{P}_f(P')}{\mathbb{P}_{f^*}(P')} \cdot \alpha^{-2} \cdot \beta^{-\ell}$$

due to $Pr(f|\mathcal{D}^*) > \beta^\ell$ and $Pr(f|\mathcal{D}^*) \geq \gamma$. According to Lemma F.1, as long as $\frac{\mathbb{P}_f(P')}{\mathbb{P}_{f^*}(P')} \leq \frac{\mathbb{P}_{\mathcal{D}^*}(W|x) - \mathbb{P}_{\mathcal{D}^*}(\hat{W}|x)}{5\alpha^{-2}\beta^{-\ell}\gamma^{-1}} := \mu$ and the M is properly choosen, we have $\left|\frac{B}{C}\right| \leq \frac{1}{5}(\mathbb{P}_{\mathcal{D}^*}(W|x) - \mathbb{P}_{\mathcal{D}^*}(\hat{W}|x))$ with probability at least $1-q$. Similarly, we can bound $|\frac{D}{C}| < \frac{1}{4}$.

Since C and D are non-negative, we have

$$\left|\frac{A}{C+D} - \frac{A}{C}\right| = \left|\frac{AD}{C^2+CD}\right| \leq \left|\frac{AD}{C^2}\right| = \left|\frac{A}{C}\right| \cdot \left|\frac{D}{C}\right|,$$

implying

$$\frac{A}{C+D} \geq \frac{A}{C} - \left|\frac{A}{C}\right| \cdot \left|\frac{D}{C}\right| \geq \frac{A}{C}\left(1 - \left|\frac{D}{C}\right|\right) \geq \frac{3}{4}\frac{A}{C}. \tag{41}$$

Similarly, we can bound

$$\frac{B}{C+D} \geq \frac{B}{C} - \left|\frac{B}{C}\right|\left|\frac{D}{C}\right| \geq -\left|\frac{B}{C}\right|\left(1 + \left|\frac{D}{C}\right|\right) \geq -\frac{5}{4}\left|\frac{B}{C}\right|. \tag{42}$$

To bound $\frac{A}{C}$, by definition we have

$$\frac{A}{C} = \mathbb{P}_{\mathcal{D}^*}(W|x) - \mathbb{P}_{\mathcal{D}^*}(\hat{W}|x) + (\alpha^2-1)\mathbb{P}_f(W|x) + (\alpha^{-2}-1)\mathbb{P}_f(W^*|x) > \mathbb{P}_{\mathcal{D}^*}(W|x) - \mathbb{P}_{\mathcal{D}^*}(\hat{W}|x) - 1 + \alpha^2. \tag{43}$$

According to (41)-(43), we have

$$\mathbb{P}_{\mathcal{D}}(W|P' \oplus x) - \mathbb{P}_{\mathcal{D}}(W^*|P' \oplus x) > \frac{3}{4}(\mathbb{P}_{\mathcal{D}^*}(W|x) - \mathbb{P}_{\mathcal{D}^*}(\hat{W}|x) - 1 + \alpha^2) - \frac{5}{4}\cdot\frac{1}{5}(\mathbb{P}_{\mathcal{D}^*}(W|x) - \mathbb{P}_{\mathcal{D}^*}(\hat{W}|x)) > \frac{1}{2}(\mathbb{P}_{\mathcal{D}^*}(W|x) - \mathbb{P}_{\mathcal{D}^*}(\hat{W}|x)) + \alpha^2 - 1.$$

We then finish the proof.

G Proof of Theorem 6.2

We first prove the upper bound of ICL prediction loss. By choosing $\mu = \frac{\mathbb{P}_{\mathcal{D}^*}(W|x) - \mathbb{P}_{\mathcal{D}^*}(\hat{W}|x)}{(1-\frac{c}{2})^{-1}\alpha^{-2}\beta^{-\ell}\gamma^{-1}}$, according to Lemma 6.1, we have

$$\mathbb{P}_{\mathcal{D}}(W|P' \oplus x) - \mathbb{P}_{\mathcal{D}}(W^*|P' \oplus x) > (1-c)(\mathbb{P}_{\mathcal{D}^*}(W|x) - \mathbb{P}_{\mathcal{D}^*}(\hat{W}|x)) + \alpha^{-2} - 1.$$

Consider that $\epsilon = \frac{2\Delta_{pre}}{c(1-c)}$. If $\mathbb{P}_{\mathcal{D}}(W|P' \oplus x) - \mathbb{P}_{\mathcal{D}}(W^*|P' \oplus x) > \epsilon$, we have

$$\begin{aligned}
\mathbb{P}_{\mathcal{D}}(W|P' \oplus x) - \mathbb{P}_{\mathcal{D}}(W^*|P' \oplus x) &> (1-c)(\mathbb{P}_{\mathcal{D}^*}(W|x) - \mathbb{P}_{\mathcal{D}^*}(\hat{W}|x)) + \alpha^{-2} - 1 \\
&> (1-c)(\mathbb{P}_{\mathcal{D}^*}(W|x) - \mathbb{P}_{\mathcal{D}^*}(\hat{W}|x)) - (1-c)^2\Delta_{\mathcal{D}^*} \\
&\geq (1-c)(\mathbb{P}_{\mathcal{D}^*}(W|x) - \mathbb{P}_{\mathcal{D}^*}(\hat{W}|x)) - (1-c)^2(\mathbb{P}_{\mathcal{D}^*}(W|x) - \mathbb{P}_{\mathcal{D}^*}(\hat{W}|x)) \\
&= (1-c)c(\mathbb{P}_{\mathcal{D}^*}(W|x) - \mathbb{P}_{\mathcal{D}^*}(\hat{W}|x)) > 2\Delta_{pre},
\end{aligned}$$

which implies that the ICL prediction is exactly the same as the ground truth and there incurs no ICL prediction loss. If $\mathbb{P}_{\mathcal{D}}(W|P' \oplus x) - \mathbb{P}_{\mathcal{D}}(W^*|P' \oplus x) \leq \epsilon$, the mismatching probability of our ICL prediction is less than $\epsilon = \frac{2\Delta_{pre}}{c(1-c)}$, indicating that our ICL prediction loss is bounded by ϵ.

Next, we prove our throughput loss bound. We have the ICL prediction loss $E[\mathbf{1}_{(\hat{W} \neq W)}] \leq \frac{2\Delta_{pre}}{c(1-c)} + \text{BER}$, where BER stands for error rate of the Bayes optimal classifier. We then have

$$\begin{aligned}
\mathbb{E}[U(W) - U(\hat{W})] =& \mathbb{E}[|U(W) - U(\hat{W}|] \\
\leq& \mathbb{E}\left[\frac{T_P\bar{N}}{8T_\sigma} \cdot |W - \hat{W}|\right] \\
\leq& \frac{T_P\bar{N}}{8T_\sigma} \cdot \left(\mathbb{E}(W - \hat{W})^2\right)^{\frac{1}{2}} \\
\leq& \frac{T_P\bar{N}}{8T_\sigma} \cdot \left(\mathbb{E}[\mathbf{1}_{(W \neq \hat{W})}\bar{W}^2]\right)^{\frac{1}{2}} \\
\leq& \frac{T_P\bar{N}}{8T_\sigma} \cdot \left(\left(\frac{2\Delta_{pre}}{c(1-c)} + \text{BER}\right)\bar{W}^2\right)^{\frac{1}{2}} \\
=& \frac{T_P\bar{N}}{8T_\sigma} \cdot \left(\frac{2\Delta_{pre}}{c(1-c)} + \cdot\text{BER}\right)^{\frac{1}{2}}\bar{W} = O\left(\frac{T_P\bar{N}}{8T_\sigma}\left(\frac{2\Delta_{pre}}{c(1-c)}\right)^{\frac{1}{2}}\bar{W}\right),
\end{aligned}$$

where the first inequality holds due to Lemma 5.4 and the second holds due to Jensen's inequality. We then finish the proof.