Task-oriented Age of Information for Remote Inference with Hybrid Language Models

Shuying Gan[1], Xijun Wang[1*], Chenyuan Feng[2*], Chao Xu[3], Howard H. Yang[4], Xiang Chen[1], and Tony Q. S. Quek[5]
[1]School of Electronics and Information Engineering, Sun Yat-sen University, Guangzhou, China
[2]Department of Communication Systems, EURECOM, Sophia Antipolis, France
[3]School of Information Engineering, Northwest A&F University, Yangling, China
[4]ZJU-UIUC Institute, Zhejiang University, Haining, China
[5]Information System and Technology Design Pillar, Singapore University of Technology and Design, Singapore
Email: ganshy7@mail2.sysu.edu.cn, wangxijun@mail.sysu.edu.cn, Chenyuan.Feng@eurecom.fr,
cxu@nwafu.edu.cn, haoyang@intl.zju.edu.cn, chenxiang@mail.sysu.edu.cn, tonyquek@sutd.edu.sg

***Abstract*—Large Language Models (LLMs) have revolutionized the field of artificial intelligence (AI) through their advanced reasoning capabilities, but their extensive parameter sets introduce significant inference latency, posing a challenge to ensure the timeliness of inference results. While Small Language Models (SLMs) offer faster inference speeds with fewer parameters, they often compromise accuracy on complex tasks. This study proposes a novel remote inference system comprising a user, a sensor, and an edge server that integrates both model types alongside a decision maker. The system dynamically determines the resolution of images transmitted by the sensor and routes inference tasks to either an SLM or LLM to optimize performance. The key objective is to minimize the Task-oriented Age of Information (TAoI) by jointly considering the accuracy and timeliness of the inference task. Due to the non-uniform transmission time and inference time, we formulate this problem as a Semi-Markov Decision Process (SMDP). By converting the SMDP to an equivalent Markov decision process, we prove that the optimal control policy follows a threshold-based structure. We further develop a relative policy iteration algorithm leveraging this threshold property. Simulation results demonstrate that our proposed optimal policy significantly outperforms baseline approaches in managing the accuracy-timeliness trade-off.**

***Index Terms*—Remote inference, task-oriented age of information, semi-Markov decision process, small language models, large language models**

I. Introduction

It is prevalent to provide artificial intelligence (AI) services in remote inference systems using status update data collected from sensors, such as in applications like intelligent transportation, industrial automation, and personal assistance [1]. Within these frameworks, the execution of inference tasks relies on the transmission of data to pre-trained neural networks, where both the precision and timeliness of inference are paramount for maintaining the quality of service. Large Language Models (LLMs), celebrated for their extensive comprehension and reasoning skills, have become prominent AI services for ensuring the accuracy of inferences [2], [3]. However, the pursuit of enhanced accuracy in LLMs has resulted in models with an enormous parameter count, such as GPT-4 and LLaMA-405B, leading to a notable increase in inference latency. In contrast, Small Language Models (SLMs), with reduced parameter sets, enable more rapid inference but may compromise accuracy, especially when dealing with intricate tasks [4], as seen with models like LLaMA-7B and ALBERT. Consequently, the effective orchestration of SLMs and LLMs within remote inference systems to achieve a balance between inference accuracy and timeliness presents a challenge that merits exploration.

Previous studies have explored methods for achieving efficient and timely inference in remote inference systems. In [5], the authors showed that inference error is not necessarily a linear function of the age of information (AoI) nor a non-increasing function of the feature length. They jointly optimized feature length selection and transmission scheduling to minimize the average inference error. Building on this work, the authors of [6] proposed a selection-from-buffer model for feature selection to reduce inference error. In [7], the focus was on minimizing remote inference error for a dynamically changing objective at the receiver. The concept of task-oriented age of information (TAoI) was introduced in [8] to quantify the timeliness of the inference tasks in a remote inference system with pre-discrimination at the transmitter. A limitation of the remote inference systems in the above works is that they were restricted to scenarios with a single network at the receiver, ignoring the impact of network architecture on inference performance. The authors of [9] proposed an online optimization framework for multi-user and multi-DNN inference services. This framework aimed to strike a balance between inference precision, latency, and resource expenditure by jointly optimizing DNN model selection and resource allocation. While [9] recognized the importance of accounting for multiple neural networks at the receiver, solely focusing on minimizing inference latency is insufficient to guarantee the timeliness of remote inference systems.

This work was supported in part by the National Key Research and Development Program of China under Grant 2024YFE0200300, in part by the National Natural Science Foundation of China under Grants 62271513 and 62271413, in part by the Research Fund under the Shaanxi Province Innovation Capability Support Program under Grant 2023KJXX-010, and in part by the Talents Special Foundation of Northwest A&F University under Grant Z1090324128.

Motivated by these limitations, we investigate a remote inference system with hybrid SLM and LLM. In particular, the system consists of a user, a sensor, and an edge server equipped with a decision maker, an SLM, and an LLM. Given that different image resolutions and model sizes result in varying transmission and inference latencies as well as accuracies, the decision maker controls the resolution of the image transmitted by the sensor and decides whether to forward it to the SLM or LLM for inference. To strike a balance between timeliness and accuracy, we employ TAoI as the performance metric, which is reduced upon successful inference and accumulates otherwise, and aim to develop an optimal control strategy that minimizes the TAoI. By modeling this dynamic control problem as a finite Semi-Markov Decision Process (SMDP) and then converting it into a Markov Decision Process (MDP) with uniform time steps, we prove that the optimal policy adheres to a threshold-based structure. Furthermore, we propose a Relative Policy Iteration (RPI) algorithm that leverages this threshold-based approach to yield the optimal control policy. Finally, simulation results verify that the proposed policy outperforms baseline strategies in terms of TAoI minimization.

The rest of this paper is organized as follows. Section II introduces the system model. In Section III, we present the SMDP formulation and design an RVI algorithm based on the threshold structure. Section IV discusses the simulation results, followed by the conclusion in Section V.

II. System Model

As shown in Fig. 1, we consider a remote inference system consisting of a user, a sensor, and an edge server. The sensor captures real-time scenes and generates images at resolutions determined by the edge server. These images are then transmitted to the edge server, which houses three key components: a decision maker, a SLM, and a LLM. The system balances precision and timeliness in executing inference tasks, such as responding to queries like "What is the current license plate number?". High-resolution images, while providing more detailed clarity beneficial for complex inference tasks, come with increased transmission latency. For instance, when a vehicle is at a distance from the sensor, a high-resolution image is crucial for accurately identifying the license plate, despite the longer transmission time. In contrast, when a vehicle is in close proximity, a low-resolution image suffices for the inference task and results in faster transmission. Upon receiving images, the decision maker decides whether the current inference task should be handled by the SLM or the LLM. The SLM offers faster processing with moderate accuracy, while the LLM provides higher accuracy at the cost of increased processing time. The selected model generates text output for the user, who provides satisfaction feedback to the decision maker. This feedback loop enables continuous refinement of the inference process.

We consider that the system is time-slotted, where each time slot lasts for a duration of τ seconds. A decision epoch of the decision maker is denoted as a discrete time step t, and each

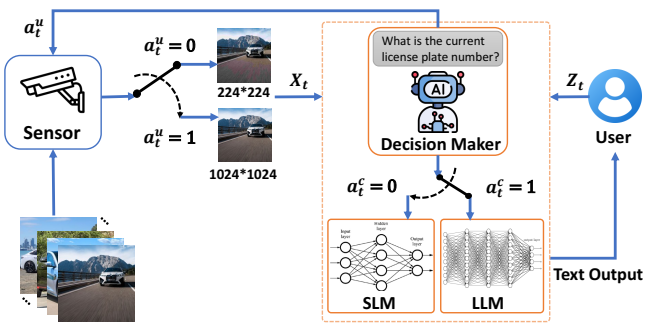


Fig. 1: An illustration of the remote inference system with hybrid SLM and LLM.

time step contains multiple time slots. At the beginning of each time step, the decision maker selects the resolution of the image to be transmitted and decides on the language model to be used. Let $a_t^u \in \{0, 1\}$ denote the resolution decision, with $a_t^u = 0$ indicating that the sensor is instructed to transmit a low-resolution image, and $a_t^u = 1$ indicating that a high-resolution image is to be sent. Based on the resolution decision a_t^u, the sensor captures a real-time scene and generates an image $X_t \in \mathcal{X}$ at the specified resolution, which is then transmitted to the edge server. We assume that the transmission process is reliable, with the transmission latency for a low-resolution image being T_1^u and for a high-resolution image being T_2^u. Note that T_1^u is always less than T_2^u. Let a_t^c denote the inference decision, where $a_t^c = 0$ signifies inference by the SLM, and $a_t^c = 1$ signifies inference by the LLM. When the edge server receives the transmitted image X_t, the decision maker sends the image and its corresponding query to the SLM or the LLM for inference according to a_t^c. Let T_1^c and T_2^c denote the inference latencies for the SLM and LLM, respectively, with $T_1^c < T_2^c$. Then, the control action vector of the decision maker at time step t is denoted by $\boldsymbol{a}_t \triangleq (a_t^u, a_t^c) \in \mathcal{A} \triangleq \{(0,0), (0,1), (1,0), (1,1)\}$, where $\mathcal{A}$ represents the set of all possible actions. Note that the duration of a time step is not constant. Specifically, let $L(\boldsymbol{a}_t)$ represent the number of time slots within time step t when action $\boldsymbol{a}_t$ is executed. This can be formulated as:

$$L(\boldsymbol{a}_t) = \begin{cases} T_1^u + T_1^c, & \text{if } \boldsymbol{a}_t = (0,0); \\ T_2^u + T_1^c, & \text{if } \boldsymbol{a}_t = (1,0); \\ T_1^u + T_2^c, & \text{if } \boldsymbol{a}_t = (0,1); \\ T_2^u + T_2^c, & \text{if } \boldsymbol{a}_t = (1,1). \end{cases} \tag{1}$$

The user, upon receiving the text output of the language model, sends feedback Z_t to the receiver, with $Z_t = 1$ indicating a correct output and $Z_t = 0$ indicating otherwise. We assume that the latency associated with transmitting the text output to the user is negligible. It is important to note that the inference accuracy is influenced not only by the size of the language model but also by the image resolution. Let p_s and q_s denote the probabilities of correct inference when a low-resolution and high-resolution image, respectively, are sent to the SLM for inference, i.e.,

$$p_s \triangleq \Pr(Z_t = 1|\mathbf{a}_t\text{= (0,0)}), \forall t, \tag{2}$$

$$q_s \triangleq \Pr(Z_t = 1|\mathbf{a}_t\text{= (1,0)}), \forall t. \tag{3}$$

Similarly, p_l and q_l are defined as the probability of correct inference when a low-resolution and high-resolution image, respectively, are transmitted to the LLM for inference, i.e.,

$$p_l \triangleq \Pr(Z_t = 1|\mathbf{a}_t\text{= (0,1)}), \forall t, \tag{4}$$

$$q_l \triangleq \Pr(Z_t = 1|\mathbf{a}_t\text{= (1,1)}), \forall t. \tag{5}$$

AoI serves as a prevalent metric for quantifying the freshness of data as perceived by the receiver [10]. However, it does not capture the utility of the information content with respect to the specific task. To bridge this gap, our remote inference system employs TAoI to measure the accuracy and timeliness of the inference task [8]. Specifically, TAoI only decreases upon the successful completion of an inference task; in other cases, it increases. Let U_t denote the time step at which the most up-to-date correct text output received by the user was generated. The TAoI at the i-th time slot of the time step t is defined as $\Delta_{t,i} = \sum_{n=U_t}^{t-1} L(\mathbf{a}_n) + i - 1$, where the first term represents the total number of time slots in the previous time steps since U_t. For ease of explanation, we represent the TAoI at the beginning of time step t as Δ_t. That is, $\Delta_t = \Delta_{t,1} = \sum_{n=U_t}^{t-1} L(\mathbf{a}_n)$. We introduce $\hat{\Delta}$ as the upper limit of the TAoI, which is assumed to be finite but can be arbitrarily large. Upon successful completion of the inference task (i.e., $Z_t = 1$), TAoI is reduced to its corresponding total latency. For instance, if a low-resolution image is transmitted, the SLM is selected for inference, and the text output is correct (i.e., $\mathbf{a}_t = (0,0)$ and $Z_t = 0$), then the TAoI resets to $T_1^u + T_1^c$. Conversely, if the inference task fails (i.e., $\mathbf{a}_t = (0,0)$ and $Z_t = 0$), TAoI increases by $T_1^u + T_1^c$. Therefore, the evolution of TAoI can be illustrated as follows:

$$\Delta_{t+1} = \begin{cases} T_1^u + T_1^c, & \mathbf{a}_t\text{= (0,0) \& } Z_t = 1; \\ T_2^u + T_1^c, & \mathbf{a}_t\text{= (1,0) \& } Z_t = 1; \\ T_1^u + T_2^c, & \mathbf{a}_t\text{= (0,1) \& } Z_t = 1; \\ T_2^u + T_2^c, & \mathbf{a}_t\text{= (1,1) \& } Z_t = 1; \\ \min\{\Delta_t + T_1^u + T_1^c, \hat{\Delta}\}, & \mathbf{a}_t\text{= (0,0) \& } Z_t = 0; \\ \min\{\Delta_t + T_2^u + T_1^c, \hat{\Delta}\}, & \mathbf{a}_t\text{= (1,0) \& } Z_t = 0; \\ \min\{\Delta_t + T_1^u + T_2^c, \hat{\Delta}\}, & \mathbf{a}_t\text{= (0,1) \& } Z_t = 0; \\ \min\{\Delta_t + T_2^u + T_2^c, \hat{\Delta}\}, & \mathbf{a}_t\text{= (1,1) \& } Z_t = 0. \end{cases} \tag{6}$$

In this study, our objective is to develop a control policy $\pi = (\mathbf{a}_1, \mathbf{a}_2, \cdots)$ that minimizes the long-term average TAoI. The dynamic control problem can be formulated as follows:

$$\min_{\pi} \limsup_{T\to\infty} \frac{\mathbb{E}\left[\sum_{t=1}^{T} \Delta_t\right]}{\mathbb{E}\left[\sum_{t=1}^{T} L(\mathbf{a}_t)\right]}. \tag{7}$$

III. SMDP Formulation and Solution

A. SMDP Formulation

Due to the non-uniform durations of time intervals between decision epochs, we reformulate the dynamic control problem (7) as the SMDP. Specifically, an SMDP is composed of a tuple $(\mathcal{S}, \mathcal{A}, t^+, \Pr(\cdot,\cdot), R(\cdot,\cdot))$, where each component is defined as follows:

TABLE I: Transition probability

$\Pr(s_{t+1}\|s_t, a_t)$	s_t	$\mathbf{a}_t$	s_{t+1}
p_s	Δ_t	$(0,0)$	$T_1^u + T_1^c$
$1-p_s$	Δ_t	$(0,0)$	$\min\{\Delta_t + T_1^u + T_1^c, \hat{\Delta}\}$
q_s	Δ_t	$(1,0)$	$T_2^u + T_1^c$
$1-q_s$	Δ_t	$(1,0)$	$\min\{\Delta_t + T_2^u + T_1^c, \hat{\Delta}\}$
p_l	Δ_t	$(0,1)$	$T_1^u + T_2^c$
$1-p_l$	Δ_t	$(0,1)$	$\min\{\Delta_t + T_1^u + T_2^c, \hat{\Delta}\}$
q_l	Δ_t	$(1,1)$	$T_2^u + T_2^c$
$1-q_l$	Δ_t	$(1,1)$	$\min\{\Delta_t + T_2^u + T_2^c, \hat{\Delta}\}$

1) State space $\mathcal{S}$: The state of the SMDP at time step t is defined as the TAoI, denoted by $s_t \triangleq \Delta_t$. Given that the TAoI is bounded by its upper limit $\hat{\Delta}$, the state space $\mathcal{S}$ is finite.

2) Action space $\mathcal{A}$: The action of the SMDP at time step t comprises a resolution decision and an inference decision made by the decision maker, denoted by $\mathbf{a}_t \triangleq (a_t^u, a_t^c)$. The action space is $\mathcal{A} \triangleq \{(0,0),(0,1),(1,0),(1,1)\}$.

3) Decision epoch t^+: The time interval $L(\mathbf{a}_t)$ between two consecutive decision epochs is determined by the action $\mathbf{a}_t$ taken at time step t, as detailed in (1).

4) Transition probability $\Pr(\cdot,\cdot)$: Let $\Pr(s_{t+1}|s_t, \mathbf{a}_t)$ denote the transition probability from the current state s_t to the next state s_{t+1} under action $\mathbf{a}_t$. According to the TAoI evolution dynamics in (6), the transition probabilities are detailed in Table I.

5) Cost function $R(\cdot,\cdot)$: We define the instantaneous cost under state s_t given action $\boldsymbol{a}_t$ as follows:

$$\begin{aligned} R(s_t, \mathbf{a}_t) = R(\Delta_t, \mathbf{a}_t) &= \sum_{i=1}^{L(\mathbf{a}_t)} \Delta_{t,i} = \sum_{i=1}^{L(\mathbf{a}_t)} \Delta_t + i - 1 \\ &= L(\mathbf{a}_t)[\Delta_t + \frac{1}{2}(L(\mathbf{a}_t) - 1)]. \end{aligned} \tag{8}$$

Given an initial system state s_1, the objective can be expressed as follows:

$$\min_{\pi} \limsup_{T\to\infty} \frac{\mathbb{E}\left[\sum_{t=1}^{T} R(s_t, \mathbf{a}_t) \mid s_1\right]}{\mathbb{E}\left[\sum_{t=1}^{T} L(\mathbf{a}_t)\right]}. \tag{9}$$

Our goal is to find a stationary deterministic optimal control policy π^* that solves the long-term average TAoI minimization problem as presented in (9). Before analyzing the stationary deterministic optimal policy for average TAoI, it is imperative to confirm the existence of such a policy. According to [11, Theorem 8.4.5], a deterministic stationary average optimal policy exists for a finite-state finite-action average-cost MDP provided that the cost function is bounded and the MDP is unichain. Thus, we need examine the two prerequisites for the existence of a deterministic stationary policy: i) First, the cost in the MDP is bounded, as the instantaneous cost is defined by the TAoI, which is capped by an upper limit $\hat{\Delta}$; ii) Second, given that the state $\hat{\Delta}$ is accessible from every other state, our Markov chain forms a single recurrent class, signifying that

the MDP is unichain. Hence, a stationary deterministic optimal policy is confirmed to exist for this dynamic control problem.

To derive the optimal control policy, we begin by converting the SMDP into an equivalent discrete-time MDP [11]. Let $\hat{\mathcal{S}}$ and $\hat{\mathcal{A}}$ denote the state and action spaces of the transformed MDP, respectively. These spaces are identical to those of the original SMDP, that is, $\hat{\mathcal{S}} = \mathcal{S}$ and $\hat{\mathcal{A}} = \mathcal{A}$. For any state $s = \Delta \in \hat{\mathcal{S}}$ and action $\boldsymbol{a} \in \hat{\mathcal{A}}$, the cost in the MDP is

$$\bar{R}(\Delta, \boldsymbol{a}) = \frac{R(\Delta, \boldsymbol{a})}{L(\boldsymbol{a})} = \Delta + \frac{1}{2}(L(\boldsymbol{a}) - 1), \tag{10}$$

and the transition probability is given by

$$\bar{p}(s'|s, \boldsymbol{a}) = \begin{cases} \frac{\epsilon}{L(\boldsymbol{a})} p(s'|s, \boldsymbol{a}), & s' \neq s \\ 1 - \frac{\epsilon}{L(\boldsymbol{a})}, & s' = s \end{cases}, \tag{11}$$

where ϵ is selected to be within the interval $(0, \min_{\boldsymbol{a}} L(\boldsymbol{a})]$. The objective is then to find a policy $\pi \in \Pi$ that minimizes the following:

$$\min_{\pi \in \Pi} \frac{1}{T} \limsup_{T \to \infty} \mathbb{E}\left[\sum_{t=1}^{T} \bar{R}(s_t, \boldsymbol{a}_t) \mid s_1\right]. \tag{12}$$

We focus on the set of deterministic stationary policies Π, where $\pi = \{\boldsymbol{a}_1, \boldsymbol{a}_2, \cdots\} \in \Pi$ such that $\boldsymbol{a}_{t_1} = \boldsymbol{a}_{t_2}$ when $s_{t_1} = s_{t_2}$ for any t_1, t_2. For simplicity, we omit the time index in the sequel. The optimal policy π^* can be derived by solving the corresponding Bellman equation. According to [12], we have:

$$V^* + V(s) = \min_{\boldsymbol{a} \in \mathcal{A}} \left\{ \bar{R}(s, \boldsymbol{a}) + \sum_{s' \in \mathcal{S}} \bar{p}(s'|s, \boldsymbol{a}) V(s') \right\}, \ \forall s \in \mathcal{S}, \tag{13}$$

where V^* represents the optimal value to (9) for all initial states, and $V(\boldsymbol{s})$ is the value function for the discrete-time MDP. The optimal policy π^* for any state $\boldsymbol{s} \in \mathcal{S}$ is given by:

$$\pi^*(s) = \arg\min_{\boldsymbol{a} \in \mathcal{A}} \left\{ \bar{R}(s, \boldsymbol{a}) + \sum_{s' \in \mathcal{S}} \bar{p}(s'|s, \boldsymbol{a}) V(s') \right\}, \ \forall s \in \mathcal{S}. \tag{14}$$

B. Structural Analysis and Optimal Policy

Our first step is to prove that the optimal policy exhibits a threshold-like structure. Based on this, we develop an RPI algorithm that exploits this threshold structure to find the optimal policy. To proceed, we present key properties of the value function, as shown in the following lemmas.

Lemma 1. *The value function $V(\Delta)$ is non-decreasing with Δ.*

Proof: See Section II-A in the online materials [13]. □

Lemma 2. *The value function $V(\Delta)$ is concave with Δ.*

Proof: See Section II-B in the online materials [13]. □

Since the value function $V(\Delta)$ is non-decreasing and concave, its slope is non-increasing and lower bounded. The lower bound of the slope of $V(\Delta)$ is given by the following lemma. Prior to that, we define an auxiliary variable l_{min} as follows:

$$l_{min} = \min\left(\frac{T_1^u + T_1^c}{p_s}, \frac{T_2^u + T_1^c}{q_s}, \frac{T_1^u + T_2^c}{p_l}, \frac{T_2^u + T_2^c}{q_l}\right). \tag{15}$$

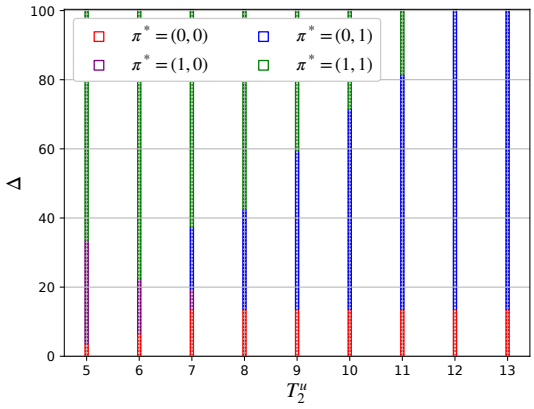


Fig. 2: Structure of the optimal policy for different T_2^u ($T_1^u = 4$, $T_1^c = 3$, $T_2^c = 4$, $p_s = 0.3$, $q_s = 0.7$, $p_l = 0.5$, $q_l = 0.8$).

Lemma 3. *For any Δ_1, $\Delta_2 \in \mathcal{S}$ with $\Delta_1 \leq \Delta_2$, we have $V(\Delta_2) - V(\Delta_1) \geq \frac{L(\hat{\mathbf{a}})}{\epsilon \hat{p}}(\Delta_2 - \Delta_1)$, where $\hat{\mathbf{a}}$ and $\hat{p}$ are given by*

$$(\hat{\mathbf{a}}, \hat{p}) = \begin{cases} ((0,0), p_s), & \text{if } l_{min} = \frac{T_1^u + T_1^c}{p_s}; \\ ((1,0), q_s), & \text{if } l_{min} = \frac{T_2^u + T_1^c}{q_s}; \\ ((0,1), p_l), & \text{if } l_{min} = \frac{T_1^u + T_2^c}{p_l}; \\ ((1,1), q_l), & \text{if } l_{min} = \frac{T_2^u + T_2^c}{q_l}. \end{cases} \tag{16}$$

Proof: See Section II-C in the online materials [13]. □

Based on Lemmas 1-3, we can derive the structure of the optimal control policy as stated in the following theorem.

Theorem 4. *For any $\Delta_1, \Delta_2 \in \mathcal{S}$ with $\Delta_1 \leq \Delta_2$, there exists a stationary deterministic optimal policy with a threshold-based structure, described as follows:*

- *When $l_{min} = \frac{T_1^u + T_1^c}{p_s}$ and $\pi^*(\Delta_1) = (0,0)$, $\pi^*(\Delta_2) = (0,0)$.*
- *When $l_{min} = \frac{T_2^u + T_1^c}{q_s}$ and $\pi^*(\Delta_1) = (1,0)$, $\pi^*(\Delta_2) = (1,0)$.*
- *When $l_{min} = \frac{T_1^u + T_2^c}{p_l}$ and $\pi^*(\Delta_1) = (0,1)$, $\pi^*(\Delta_2) = (0,1)$.*
- *When $l_{min} = \frac{T_2^u + T_2^c}{q_l}$ and $\pi^*(\Delta_1) = (1,1)$, $\pi^*(\Delta_2) = (1,1)$.*

Proof: Please refer to Appendix A. □

Theorem 4 shows the existence of a threshold structure within the optimal policy across four different cases. It is further verified by Fig. 2, which shows that the structure of the optimal policy corresponds to case 4 when $T_2^u \leq 11$ and to case 3 when $T_2^u > 11$. Based on this threshold structure, we propose the RPI algorithm, as outlined in Algorithm 1. Specifically, if the condition outlined in Theorem 4 is satisfied, the optimal policy can be determined directly within lines 5-12 of Algorithm 1 without the need to search through all possible actions. This significantly reduces the computational complexity of the algorithm.

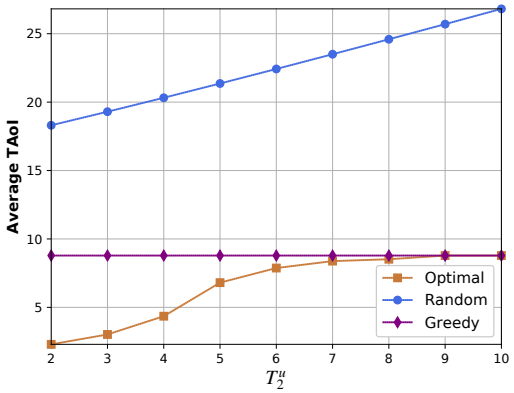
Algorithm 1 RPI Algorithm Based on the Threshold Structure

Initialization: Set $\pi_0^*(s) = 0$ for all $s \in S$, select a reference state $s^\dagger$, and set $k = 0$.
Policy Evaluation: Given π_k^* and $V_k(s^\dagger)$, compute V_k^* and $V_k(s)$ according to $V_k^* + V_k(s) = \bar{R}(s, \pi_k^*(s)) + \sum_{s' \in \mathcal{S}} \bar{p}(s'|s, \pi_k^*(s)) V_k(s')$.
Policy Improvement Based on the Threshold Structure: Compute a new policy π_k^* for each $s \in S$ as follows:
for $s \in S$ **do**
  if $l_{min} = \frac{T_1^u + T_1^c}{p_s}$ and $\pi_{k+1}^*(s-1) = (0,0)$ **then**
    $\pi_{k+1}^*(s) = (0,0)$;
  else if $l_{min} = \frac{T_2^u + T_1^c}{q_s}$ and $\pi_{k+1}^*(s-1) = (1,0)$ **then**
    $\pi_{k+1}^*(s) = (1,0)$;
  else if $l_{min} = \frac{T_1^u + T_2^c}{p_l}$ and $\pi_{k+1}^*(s-1) = (0,1)$ **then**
    $\pi_{k+1}^*(s) = (0,1)$;
  else if $l_{min} = \frac{T_2^u + T_2^c}{q_l}$ and $\pi_{k+1}^*(s-1) = (1,1)$ **then**
    $\pi_{k+1}^*(s) = (1,1)$;
  else
    $\pi_{k+1}^*(s) = \arg\min_{\mathbf{a} \in \mathcal{A}} \{\bar{R}(s, \pi_k^*(s)) + \sum_{s' \in \mathcal{S}} \bar{p}(s'|s, \pi_k^*(s)) V_k(s')\}$;
  end if
end for
Let $k = k + 1$ and go to step 2 until $\pi_k^*(s) = \pi_{k+1}^*(s)$.
Return: The optimal policy π^*.

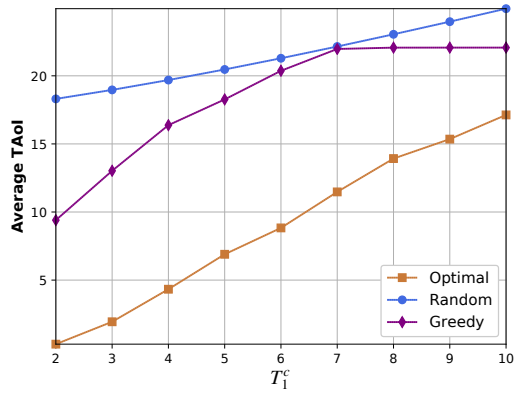
IV. Simulation Results

In this section, we conduct extensive simulations to evaluate the performance the optimal policy. We compare it against two benchmark policies, i.e., the random policy and the greedy policy. Under the random policy, the decision maker randomly selects actions at each decision epoch. In the greedy policy, the decision maker chooses the action that minimizes the expected post-action TAoI at each time step. The expected post-action TAoI is defined as the expected TAoI after the corresponding action taken in time step t. For instance, the expected post-action TAoI of action $(0,0)$ is given by $(1-p_s)(\Delta + T_1^u + T_1^c) + p_s(T_1^u + T_1^c)$. The simulation parameters T_1^u, T_2^u, T_1^c, and T_2^c are set such that $T_1^u < T_2^u$ and $T_1^c < T_2^c$. The inference accuracy for both SLM and LLM varies between 0.05 and 0.99 [14].

Fig. 3 compares the average TAoI between the optimal policy and the two baseline policies with respect to the transmission latency T_2^u and the inference latency T_1^c. While the optimal policy's average TAoI generally increases with both latency parameters, its behavior differs markedly between the two cases. For transmission latency T_2^u, shown in Figure 3(a), the average TAoI plateaus once T_2^u exceeds 9, as the optimal policy adaptively avoids high-resolution image selection, making further increases in T_2^u inconsequential. In contrast, Figure 3(b) shows that the average TAoI continues to rise with inference latency T_1^c without stabilizing, since T_1^c remains below T_2^c and thus continues to influence the system's performance through the optimal policy's decision-making process. Moreover, as shown in Fig. 3(a), when T_2^u is large, the optimal policy coincides with the greedy policy, which always selects action $(0,0)$ in this setup. As T_2^u increases, the optimal policy favors transmitting low-resolution images. Also, given that T_1^c is substantially lower than T_2^c, the potential benefits of LLM processing become outweighed by its latency costs. These combined effects drive the optimal policy to naturally align with the greedy policy.



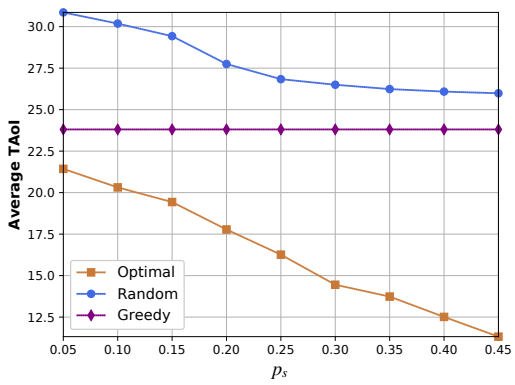
(a) Average TAoI versus T_2^u ($T_1^u = 1$, $T_1^c = 2$, $T_2^c = 11$).



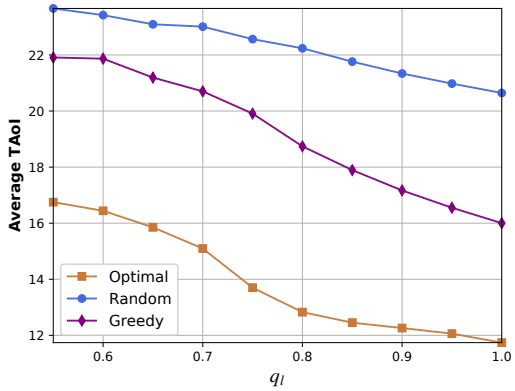
(b) Average TAoI versus T_1^c ($T_1^u = 1$, $T_2^u = 2$, $T_2^c = 11$).

Fig. 3: Average TAoI versus T_2^u or T_1^c ($p_s = 0.4$, $q_s = 0.5$, $p_l = 0.6$, $q_l = 0.8$).

In Fig. 4, the average TAoI of the optimal policy and the two baselines are compared with respect to model accuracy parameters p_s or q_l. As shown in Fig. 4(a) and Fig. 4(b), we can see that the optimal policy consistently achieves lower average TAoI compared to baseline policies. Moreover, the average TAoI of the optimal policy decreases with the increase of p_s or q_l, which indicates that, enhanced model accuracy, regardless of resolution or model size, is beneficial for the success of inference tasks. The impact of these parameters, however, manifests differently. Fig. 4(a) shows that as p_s increases, the average TAoI of the optimal policy exhibits a sharp decrease. This pronounced improvement occurs because action $(0,0)$, which offers the lowest latency, becomes increasingly favored by the optimal policy as its inference accuracy improves. In contrast, Figure 4(b) shows that while increases in q_l also reduce the average TAoI, this reduction occurs more gradually and diminishes at higher values of q_l, suggesting a



(a) Average TAoI versus p_s ($q_s = 0.5$, $p_l = 0.6$, $q_l = 0.8$).



(b) Average TAoI versus q_l ($p_s = 0.3$, $q_s = 0.4$, $p_l = 0.5$).

Fig. 4: Average TAoI versus p_s or q_l ($T_1^u = 3$, $T_2^u = 4$, $T_1^c = 8$, $T_2^c = 10$).

point of diminishing returns in the accuracy-latency trade-off.

V. Conclusions

In this paper, we introduced a novel remote inference system that combines SLM and LLM to optimize both accuracy and timeliness. We developed a dynamic control policy that minimizes the TAoI through joint optimization of resolution and inference decisions. By formulating the control problem as an infinite-horizon SMDP and transforming it into an equivalent MDP, we proved that the optimal control policy follows a threshold structure. Building on this insight, we developed a RPI algorithm that leverages this threshold structure to efficiently determine the optimal policy while minimizing computational overhead. Our extensive simulation results demonstrated the superiority of our proposed approach, with the optimal policy consistently achieving lower average TAoI compared to existing benchmark policies.

Appendix

A. Proof of Theorem 1

First, we define $Q'(\Delta_2, \Delta_1, \boldsymbol{a}) = Q(\Delta_2, \boldsymbol{a}) - Q(\Delta_1, \boldsymbol{a})$ for convenience. For any Δ_1, $\Delta_2 \in \mathcal{S}$ with $\Delta_1 \leq \Delta_2$, we have

$$\begin{aligned} &Q'(\Delta_2, \Delta_1, \hat{\boldsymbol{a}}) - (V(\Delta_2) - V(\Delta_1)) \\ =&\Delta_2 - \Delta_1 - \frac{\epsilon}{L(\hat{\boldsymbol{a}})}(V(\Delta_2) - V(\Delta_1)) \\ &+ \frac{\epsilon(1-p)}{L(\hat{\boldsymbol{a}})}(V(\Delta_2 + L(\hat{\boldsymbol{a}})) - V(\Delta_1 + L(\hat{\boldsymbol{a}}))). \end{aligned} \tag{17}$$

Given that the concavity of $V(s)$ is established in Lemma 2, it follows that $V(\Delta_2 + L(\hat{\boldsymbol{a}})) - V(\Delta_1 + L(\hat{\boldsymbol{a}})) \leq V(\Delta_2) - V(\Delta_1))$. Then, we can get that

$$\begin{aligned} &Q'(\Delta_2, \Delta_1, \hat{\boldsymbol{a}}) - (V(\Delta_2) - V(\Delta_1)) \\ &\leq \Delta_2 - \Delta_1 + \frac{\epsilon(1-p)}{L(\hat{\boldsymbol{a}})}(V(\Delta_2) - V(\Delta_1)) \\ &\quad - \frac{\epsilon}{L(\hat{\boldsymbol{a}})}(V(\Delta_2) - V(\Delta_1)) \\ &= \Delta_2 - \Delta_1 - \frac{\epsilon p}{L(\hat{\boldsymbol{a}})}(V(\Delta_2) - V(\Delta_1)). \end{aligned} \tag{18}$$

As shown in Lemma 3, we have $V(\Delta_2) - V(\Delta_1) \geq \frac{L(\hat{\boldsymbol{a}})}{\epsilon p}(\Delta_2 - \Delta_1)$. This implies that $Q'(\Delta_2, \Delta_1) - (V(\Delta_2) - V(\Delta_1)) \leq 0$.

Next, we prove the threshold structure of the optimal policy. Suppose $\Delta_2 \geq \Delta_1$ and $\pi^*(\Delta_1) = \hat{\boldsymbol{a}}$, we have $V(\Delta_1) = Q(\Delta_1, \hat{\boldsymbol{a}})$, i.e., $V(\Delta_1) = Q(\Delta_1, \hat{\boldsymbol{a}})$. It is straightforward to obtain $V(\Delta_2) \geq Q(\Delta_2, \hat{\boldsymbol{a}})$, since $V(\Delta_2) - V(\Delta_1) \geq Q(\Delta_2, \hat{\boldsymbol{a}}) - Q(\Delta_1, \hat{\boldsymbol{a}})$. Moreover, since the value function is a minimum of two state-action value functions, we have $V(\Delta_2) \leq Q(\Delta_2, \hat{\boldsymbol{a}})$. Therefore, we can conclude that $V(\Delta_2) = Q(\Delta_2, \hat{\boldsymbol{a}})$ and that $\pi^*(\Delta_2) = \hat{\boldsymbol{a}}$. This concludes the proof.

References

[1] Z. Chen *et al.*, "Enabling Mobile AI Agent in 6G Era: Architecture and Key Technologies," *IEEE Netw.*, vol. 38, no. 5, pp. 66–75, 2024.

[2] S. Guo *et al.*, "Semantic importance-aware communications using pre-trained language models," *IEEE Commun. Lett.*, vol. 27, no. 9, pp. 2328–2332, 2023.

[3] M. Lan *et al.*, "LLM4QA: Leveraging Large Language Model for Efficient Knowledge Graph Reasoning with SPARQL Query," *Journal of Advances in Information Technology*, vol. 15, no. 10, 2024.

[4] R. Eldan and Y. Li, "Tinystories: How small can language models be and still speak coherent english?" 2023. [Online]. Available: https://arxiv.org/abs/2305.07759

[5] M. K. C. Shisher *et al.*, "Learning and Communications Co-Design for Remote Inference Systems: Feature Length Selection and Transmission Scheduling," *IEEE J. Sel. Areas Inf. Theory*, vol. 4, pp. 524–538, 2023.

[6] ——, "Timely Communications for Remote Inference," *IEEE/ACM Trans. Netw.*, vol. 32, no. 5, pp. 3824–3839, 2024.

[7] C. Ari *et al.*, "Goal-Oriented Communications for Remote Inference Under Two-Way Delay with Memory," in *Proc. ISIT*, Athens, Greece, Jul. 2024, pp. 1179–1184.

[8] G. Shuying *et al.*, "Task-oriented age of information for remote monitoring systems," *arXiv Preprint: 2411.00319*, 2024.

[9] K. Zhao *et al.*, "EdgeAdaptor: Online Configuration Adaption, Model Selection and Resource Provisioning for Edge DNN Inference Serving at Scale," *IEEE Trans. Mobile Comput.*, vol. 22, no. 10, pp. 5870–5886, 2023.

[10] R. D. Yates, *et al.*, "Age of Information: An Introduction and Survey," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 5, pp. 1183–1210, 2021.

[11] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming (Wiley Series in Probability and Statistics)*. Hoboken, NJ, USA: Wiley, 2005.

[12] P. Bertsekas, Dimitri, *Dynamic Programming and Optimal Control-II*, 3rd ed. Belmont, MA, USA: Athena Sci., 2007, vol. 2.

[13] S. Gan *et al.*, "Supplementary Materials of TAoI for Inference Systems," https://github.com/ganshuying/SLM-LLM/blob/master/Supplementary-Materials-of-SLM-LLM.pdf.

[14] J. Li *et al.*, "BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models," in *Proc. PMLR*, Honolulu, Hawaii, USA, Jul. 2023, pp. 19 730–19 742.