Learning and Communications Co-Design for Remote Inference Systems: Feature Length Selection and Transmission Scheduling

Md Kamran Chowdhury Shisher, *Member, IEEE*, Bo Ji, *Senior Member, IEEE*, I-Hong Hou, *Senior Member, IEEE*, and Yin Sun, *Senior Member, IEEE*

Digital Object Identifier 10.1109/JSAIT.2023.3322620

***Abstract*—In this paper, we consider a remote inference system, where a neural network is used to infer a time-varying target (e.g., robot movement), based on features (e.g., video clips) that are progressively received from a sensing node (e.g., a camera). Each feature is a temporal sequence of sensory data. The inference error is determined by (i) the timeliness and (ii) the sequence length of the feature, where we use Age of Information (AoI) as a metric for timeliness. While a longer feature can typically provide better inference performance, it often requires more channel resources for sending the feature. To minimize the time-averaged inference error, we study a learning and communication codesign problem that jointly optimizes feature length selection and transmission scheduling. When there is a single sensorpredictor pair and a single channel, we develop low-complexity optimal co-designs for both the cases of time-invariant and timevariant feature length. When there are multiple sensor-predictor pairs and multiple channels, the co-design problem becomes a restless multi-arm multi-action bandit problem that is PSPACEhard. For this setting, we design a low-complexity algorithm to solve the problem. Trace-driven evaluations demonstrate the potential of these co-designs to reduce inference error by up to 10000 times.**

***Index Terms*—Remote inference, transmission scheduling, age of information, restless multi-armed bandit.**

I. Introduction

THE ADVANCEMENT of communication technologies and artificial intelligence has engendered the demand for remote inference in various applications, such as autonomous vehicles, health monitoring, industrial control systems, and robotic systems [1], [2], [3], [4]. For instance, accurate prediction of the robotic state during remote robotic surgery is time-critical. The remote inference problem can be tackled by using a neural network that is trained to predict a timevarying target (e.g., robot movement) based on features (e.g., video clips) sent from a remote sensing node (e.g., a camera). Each feature is a temporal sequence of the sensory output and the length of the temporal sequence is called *feature length*.

Due to data processing time, transmission errors, and transmission delay, the features delivered to the neural predictor may not be fresh, which can significantly affect the inference accuracy. To measure the freshness of the delivered features, we use the *age of information (AoI)* metric, which was first introduced in [5]. Let $U(t)$ be the generation time of the most recently delivered feature sequence. Then, AoI is the time difference between the generation time $U(t)$ and the current time t, denoted by $(t) := t-U(t)$. Recent studies [6], [7] have shown that the inference error is a function of AoI for a given feature length, but this function is *not necessarily monotonic*. Moreover, simulation results in [6] suggest that AoI-aware remote inference, wherein both the feature and its AoI are fed to the neural network, can achieve superior performance than AoI-agnostic remote inference that omits the provision of AoI to the neural network. Hence, the AoI (t) can provide useful information for reducing the inference error.

Additionally, the performance of remote inference depends on the sequence length of the feature. Longer feature sequences can carry more information about the target, resulting in the reduction of inference errors. Though a longer feature can provide better training and inference performance, it often requires more communication resources. For example, a longer feature may require a longer transmission time and may end up being stale when delivered, thus resulting in worse

Manuscript received 24 March 2023; revised 18 August 2023; accepted 30 September 2023. Date of publication 6 October 2023; date of current version 23 October 2023. The work of Md Kamran Chowdhury Shisher and Yin Sun was supported in part by the National Science Foundation under Grant CNS-2239677, and in part by the Army Research Office under Grant W911NF-21-1-0244. The work of Bo Ji was supported in part by the National Science Foundation under Grant CNS-2112694 and Grant CNS-2106427. The work of I-Hong Hou was supported in part by the National Science Foundation under Award ECCS-2127721; in part by the U.S. Army Research Laboratory and the U.S. Army Research Office under Grant W911NF-22-1-0151; and in part by the Office of Naval Research under Contract N00014-21-1-2385. *(Corresponding author: Md Kamran Chowdhury Shisher.)*

Md Kamran Chowdhury Shisher and Yin Sun are with the Department of Electrical and Computer Engineering, Auburn University, Auburn, AL 36849 USA (e-mail: mzs0153@auburn.edu; yzs0078@auburn.edu). Bo Ji is with the Department of Computer Science, Virginia Tech, Blacksburg, VA 24061 USA (e-mail: boji@vt.edu). I-Hong Hou is with the Department of Electrical and Computer Engineering, Texas A&M University at College Station, College Station, TX 77843 USA (e-mail: ihou@tamu.edu).

This article has supplementary downloadable material available at https://doi.org/10.1109/JSAIT.2023.3322620, provided by the authors.

inference performance. Hence, it is necessary to study a learning and communications co-design problem that jointly controls the timeliness and the length of the feature sequences. The contributions of this paper are summarized as follows:

- In [7], it was demonstrated that the inference error is a function of the AoI, whereas the function is not necessarily monotonic. The current paper further investigates the impact of feature length on inference error. Our information-theoretic and experimental analysis show that the inference error is a non-increasing function of the feature length (See Figs. 2(a)-3(a), and Lemma 1).
- We propose a novel learning and communications codesign framework (see Section II). In this framework, we adopted the "selection-from-buffer" model proposed in [7], which is more general than the popular "generateat-will" model that was proposed in [8] and named in [9]. In addition, we consider both time-invariant and time-variant feature length. Earlier studies, for example [7], [10], did not consider time-variant feature length. • For a single sensor-predictor pair and a single channel, this paper jointly optimizes feature length selection and transmission scheduling to minimize the timeaveraged inference error. This joint optimization is formulated as an infinite time-horizon average-cost *semiMarkov decision process* (SMDP). Such problems often lack analytical solutions or closed-form expressions. Nevertheless, we are able to derive a closed-form expression for an optimal scheduling policy in the case of time-invariant feature length (Theorem 1). The optimal scheduling time strategy is a *threshold-based policy*. Our threshold-based scheduling approach differs significantly from previous threshold-based policies in, e.g., [7], [11], [12], [13], [14], because our threshold function depends on both the AoI value and the feature length, while prior threshold functions rely solely on the AoI value. In addition, our threshold function is *not necessarily monotonic* with AoI. This is a significant difference with prior studies [11], [12], [13], [14].
- We provide an optimal policy for the case of timevariant feature length. Specifically, Theorem 2 presents the Bellman equation for the average-cost SMDP with time-variant feature length. The Bellman equation can be solved by applying either relative value iteration or policy iteration algorithms [15, Sec. 11.4.4]. Given the complexity associated with converting the average-cost SMDP into a Markov Decision Process (MDP) suitable for relative value iteration, we opt for the alternative: using the *policy iteration* algorithm to solve our averagecost SMDP. By leveraging specific structural properties of the SMDP, we can simplify the policy iteration algorithm to reduce its computational complexity. The simplified policy iteration algorithm is outlined in Algorithm 1 and Algorithm 2.
- Furthermore, we investigate the learning and communications co-design problem for multiple sensor-predictor pairs and multiple channels. This problem is a restless multi-armed, multi-action bandit problem that is known to be PSPACE-hard [16]. Moreover, proving indexability condition relating to Whittle index policy [17] for our problem is fundamentally difficult. To this end, we propose a new scheduling policy named "Net Gain Maximization" that does not need to satisfy the indexability condition (Algorithm 4).
- Numerical evaluations demonstrate that our policies for the single source case can achieve up to 10000 times performance gain compared to periodic updating and zero-wait policy (see Figs. 5-6). Furthermore, our proposed multiple source policy outperforms the maximum age-first policy (see Fig. 7) and is close to a lower bound (see Fig. 8).

A. Related Works

The age of information (AoI) has emerged as a popular metric for analyzing and optimizing communication networks [18], [19], control systems [13], [20], remote estimation [12], [21], and remote inference [6], [7]. As surveyed in [22], several studies have investigated sampling and scheduling policies for minimizing linear and nonlinear functions of AoI [7], [9], [11], [13], [14], [18], [19], [23], [24], [25], [26], [27], [28], [29]. In most previous works [9], [11], [13], [14], [18], [19], [23], [24], [25], [26], [27], [28], [29], monotonic AoI penalty functions are considered. However, in a recent study [7], it is demonstrated that the monotonic assumption is not always true for remote inference. In contrast, the inference error is a function of AoI, but the function is not necessarily monotonic. The present paper further investigates the impact of feature length on the inference error and jointly optimizes AoI and feature length.

In recent years, researchers have increasingly employed information-theoretic metrics to evaluate information freshness [6], [7], [11], [30], [31], [32], [33], [34]. In [11], [30], [31], the authors utilized Shannon's mutual information to quantify the amount of information carried by received data messages about the current source value, and used Shannon's conditional entropy to measure the uncertainty about the current source value after receiving these messages. These metrics were demonstrated to be monotonic functions of the AoI when the source follows a

time-homogeneous Markov chain [11], [31]. Built upon these findings, the authors of [34] extended this framework to include hidden Markov model. Furthermore, a Shannon's conditional entropy term $H_{\text{Shannon}}(Y_t|X_{t-(t)} = x)$ was used in [32], [33] to quantify information uncertainty. However, a gap still existed between these information-theoretic metrics and the performance of real-time applications such as remote estimation or remote inference. In our recent works [6], [7], [35] and the present paper, we have bridged this gap by using a generalized conditional entropy associated with a loss function L, called L-conditional entropy, to measure (or approximate) training and inference errors in remote inference, as well as the estimation error in remote estimation. For example, when the loss function $L(y,\hat{y})$ is chosen as a quadratic function [redacted], the L-conditional entropy $H_L(Y_t|X_{t-(t)}) = \min_{\phi} \mathsf{E}[(Y_t - \phi(X_{t-(t)}))^2]$ is exactly the minimum mean squared estimation error in remote estimation. This approach allows us to analyze how the AoI (t) affects inference and estimation errors directly, instead of relying on information-theoretic metrics as intermediaries for assessing application performance. It is worth noting that Shannon's conditional entropy is a special case of L-conditional entropy, corresponding to the inference and estimation errors for softmax regression and maximum likelihood estimation, as discussed in Section II.

The optimization of linear and non-linear functions of AoI for multiple source scheduling can be formulated as a restless multi-armed bandit problem [7], [14], [36], [37], [38]. Whittle, in his seminal work [17], proposed an index-based policy to address restless multi-armed bandit (RMAB) problems with binary actions. Our multiple source scheduling problem is a RMAB problem with multiple actions. An extension of the Whittle index policy for multiple actions was provided in [39], but it requires to satisfy a complicated *indexability* condition. In [10], the authors considered joint feature length selection and transmission scheduling, where the penalty function was assumed to be non-decreasing in the AoI, the feature length is time-invariant, and there is only one communication channel. Under these assumptions, [10] established the indexability condition and developed a Whittle Index policy. Compared to [10], our work could handle both monotonic and non-monotonic AoI penalty functions, both time-invariant and time-variant feature lengths, and both one and multiple communication channels.

Because of (i) the time-variant feature length and nonmonotonic AoI penalty function and (ii) the fact that there exist multiple transmission actions, we could not utilize the Whittle index theory to establish indexability for our multiple source scheduling problem. To address this challenge, we propose a new "Net Gain Maximization" algorithm (Algorithm 4) for multi-source feature length selection and transmission scheduling, which does not require indexability. During the revision of this paper, we found a related study [33], where the authors introduced a similar gain index-based policy for a RMAB problem with two actions: to send or not to send. The "Net Gain Maximization" algorithm that we propose is more general than the gain index-based policy in [33] due to its capacity to accommodate more than two actions in the RMAB.

II. System Model and Scheduling Policy

We consider a remote inference system composed of a sensor, a transmitter, and a receiver, as illustrated in Fig. 1. The sensor observes a time-varying target $Y_t \in \mathsf{Y}$ and feeds its measurement $V_t \in \mathsf{V}$ to the transmitter. The transmitter generates features from the sensory outputs and progressively transmits the features to the receiver through a communication channel. Within the receiver, a neural network infers the time-varying target based on the received features.

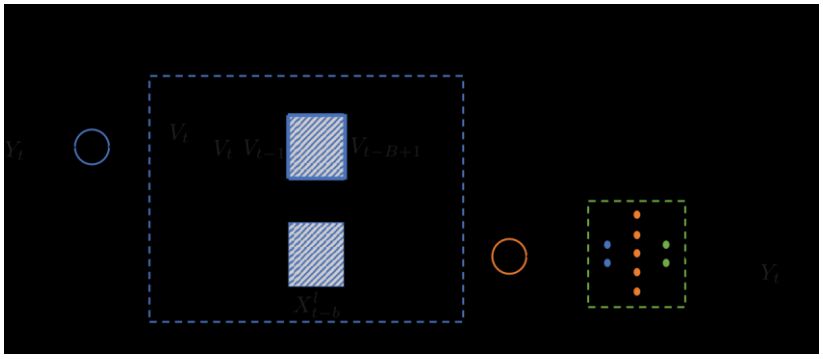


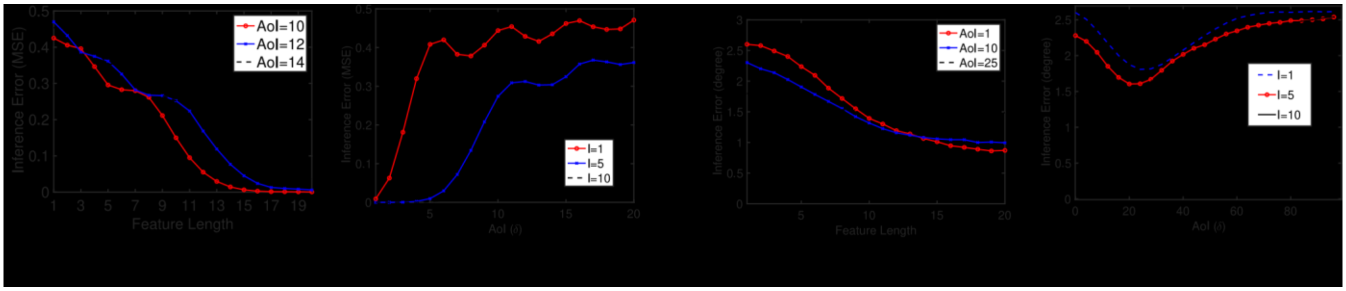
Fig. 1. A remote inference system, where $X_{t-b}^{l} := (V_{t-b}, V_{t-b-1}, \ldots, V_{t-b-l+1})$ is a feature with sequence length l.

A. System Model

The system is time-slotted and starts to operate at time slot $t = 0$. At every time slot t, the transmitter appends the sensory output $V_t \in \mathsf{V}$ to a buffer that stores the B most recent sensory outputs $(V_t, V_{t-1}, \ldots, V_{t-B+1})$; meanwhile, the

oldest output V_{t-B} is removed from the buffer. We assume that the buffer is full initially, containing B signal values

buffer remains consistently full at any time$(V^0,V^1,\ldots,V_{-B+1})$



at time $t = 0$. This ensures that the$t$.[1] The transmit-

Xter progressively generates a feature$_{tl-b}$:= $(V_{t-b},\ldots,V_{t-b-l+1}$ ∈ V $X$is a temporal sequence$_t^l$$_{-b}$, where each featureV$_l$ is the

) l

of sensory outputs taken from the buffer such that set of all l-tuples that take values from V, $1 \le l \le B$, and $0 \le b \le B-l$. For ease of presentation, the temporal sequence length l of feature X_{tl-b} is called *feature length* and the start-

ing position*position*. If the channel is idle in time slot$b$ of feature X_{tl-b} in the buffer is called$t$, the transmitter*feature* cation delays and channel errors, the feature is not instantlycan submit the feature X^l_{t-b} to the channel. Due to communi-

received. The most recently received feature is denoted as $X_{tl-\delta} = (V_{t-\delta},V_{t-\delta-1},\ldots,V_{t-\delta-l+1})$, where the latest obser-

vationcall δ the$V_{t-\delta}$*age of information (AoI)*in feature $X_{tl-\delta}$ is generatedwhich represents the dif-δ time slots ago. We

ference between the time stamps of the target Y_t and the latest

observation $V_{t-\delta}$ in feature X_{tl}-trained neural networks, eachδ. The receiver consists of B associated with a specific

feature length $l = 1,2,\ldots,B$. The neural network associated with feature length l takes

generates an [redacted] outputthe AoI δ ∈

Z+ and the feature$a$ $X^{tl}_{-\delta}$ A∈, where the neuralV^l as inputs and

network is represented by the function $\phi_l : \mathbb{Z}^+ \times \mathcal{V}^l \to \mathcal{A}$. The performance of the neural network is measured by a loss function $L : \mathcal{Y} \times \mathcal{A} \to \mathbb{R}$, where $L(y,a)$ indicates the incurred loss if the output $a \in \mathcal{A}$ is used for inference when $Y_t = y$. The loss function L is determined by the purpose of the application. For example, in softmax regression (i.e., neural network based maximum likelihood classification), the output $a = Q_Y$ is a distribution of Y_t and the loss function $L_{\log}(y,Q_Y) = -\log Q_Y(y)$ is the negative log-likelihood of the value $Y_t = y$. In neural network based mean-squared

Fig. 2. Performance of wireless channel state information prediction: (a) Inference error Vs. Feature length and (b) Inference error Vs. AoI.

estimation, a quadratic loss function $L_2(y,\hat{y}) = \|y - \hat{y}\|^2_2$ is used, where the action $a = \hat{y}$ is an estimate of the target value $Y_t = y$ and $\|y\|_2$ is the Euclidean norm of the vector y.

B. Inference Error

We assume that $\{(Y_t,X_t^l),t \in \mathbb{Z}\}$ is a stationary process for every l. Given AoI δ and feature length l, the expected inference error is a function of δ and l, given by

err$_{\text{inference}}$[redacted],   (1)

[1] This assumption does not introduce any loss of generality. If the buffer is no full at time $t = 0$, it would not affect our results.

[2] https://github.com/Kamran0153/Channel-State-Information-Prediction

where $P_{Y_t,X_{t-\delta}}$ is the joint distribution of the label Y_t and feature $X_{t-\delta}$ during online inference and the function ϕ_l represents any trained neural network that maps from $\mathbb{Z}^+ \times \mathbb{V}^l$ to $\mathbb{A}$.

The inference error $\text{err}_{\text{inference}}(\delta, l)$ can be evaluated through machine learning experiments.

In this paper, we conduct two experiments: (i) wireless channel state information (CSI) prediction and (ii) actuator states prediction in the OpenAI CartPole-v1 task [40]. Detailed information regarding the experimental setup for both experiments can be found in Appendix A of the supplementary material. The code for these experiments is available in GitHub repositories.[23]

The experimental results, presented in Figs. 2(a)-3(a), demonstrate that the inference error decreases with respect to feature length. Moreover, Figs. 2(b)-3(b) illustrate that the inference error is not necessarily a monotonic function of AoI. These findings align with machine learning experiments conducted in [6], [7], [35]. Collectively, the results from this paper and those in [6], [7], [35] indicate that longer feature lengths can enhance inference accuracy and fresher features are not always better than stale features in remote inference.

C. Feature Length Selection and Transmission Scheduling Policy

Because (i) fresh feature is not always better than stale feature and (ii) longer feature can improve inference error, we adopted "selection-from-buffer" model, which is recently proposed in [7]. In contrast to the "generate-at-will" model [8], [9], where the transmitter can only select the most recent sensory output V_t, the "selection-from-buffer" model offers greater flexibility by allowing the transmitter to pick multiple sensory outputs (which can be stale or fresh). In other words, "selection-from-buffer" model allows the transmitter to choose feature position b and feature length l under the constraints $1 \le l \le B-1$ and $0 \le b \le B-l$. Feature length selection represents a trade-off between learning and communications: A longer feature can provide better learning performance (see Figs. 2-3), whereas it requires more channel resources (e.g., more time slots or more frequency resources) for sending the feature. This motivated us to study a learningcommunication co-design problem that jointly optimizes the feature length, feature position, and transmission scheduling.

Fig. 3. Performance of actuator state prediction in the OpenAI CartPole-v1 task under mechanical response delay: (a) Inference error Vs. Feature length and (b) Inference error Vs. AoI.

The feature length and feature position may vary across the features sent over time. Feature transmissions over the channel are non-preemptive: the channel must finish sending the current feature, before becoming available to transmit the next feature. Suppose that the $(V_{S_i-b_i}, V_{S_i-b_i-1}, \ldots, V_{S_i-b_i-l_i+1})$ is submitted to the channel i-th feature $X_{S_i-b_i}^{l_i} =$ at time slot $t = S_i$, where l_i is its feature length and b_i is its feature position such that $1 \le l_i \le B$ and $0 \le b_i \le B - l_i$.

It takes $T_i(l_i) \ge 1$ time slots to send the i-th feature over the channel. The i-th feature is delivered to the receiver at time slot $D_i = S_i + T_i(l_i)$, where $S_i < D_i \le S_{i+1}$. The feature transmission time $T_i(l_i)$ depends on the feature length l_i. Due to time-varying channel conditions, we assume that, given feature length $l_i = l$, the $T_i(l)$'s are i.i.d. random variables, with a finite mean $1 \le \mathbb{E}[T_i(l)] < \infty$. Once a feature is delivered, an acknowledgment (ACK) is sent back to the transmitter, notifying that the channel has become idle.

In time slot t, the $i(t)$-th feature $X_{S_{i(t)}-b_{i(t)}}^{l_{i(t)}}$ is the most recently received feature, where $i(t) = \max_i\{D_i \le t\}$. The receiver feeds the feature $X_{S_{i(t)}-b_{i(t)}}^{l}$ to the neural network to infer Y_t. We define *age of information (AoI)* $\Delta(t)$ is defined as the difference between the time-stamp of the freshest sensory output $V_{S_{i(t)}-b_{i(t)}}^{l_{i(t)}}$ in feature $X_{S_{i(t)}-b_{i(t)}}$ and the current time t, i.e.,

$$\Delta(t) := t - \max_i\{S_i - b_i : D_i \le t\}. \tag{2}$$

Because $D_i < D_{i+1}$, it holds that

$$\Delta(t) = t - S_i + b_i, \text{ if } D_i \le t < D_{i+1}. \tag{3}$$

The initial state of the system is assumed to be $S_0 = 0, l_0 = 1, b_0 = 0, D_0 = T_0(l_0)$, and $\Delta(0)$ is a finite constant.

[3] https://github.com/Kamran0153/Impact-of-Data-Freshness-in-Learning

Let $\pi = ((S_1,b_1,l_1),(S_2,b_2,l_2),\ldots)$ represent a scheduling policy and denote the set of all the causal scheduling policies that satisfy the following conditions: (i) the scheduling time S_i, the feature position b_i, and the feature length l_i are decided based on the current and the historical information available at the transmitter such that $1 \le l_i \le B$ and $0 \le b_i \le B - l_i$ and (ii) the scheduler has access to the inference error function $\text{err}_{\text{inference}}(\cdot)$ and the distribution of $T_i(l)$ for each $l = 1,2,\ldots,B$. We use $_{\text{inv}} \subset$ to denote the set of causal scheduling policies with time-invariant feature length, defined as

$$_{\text{inv}} : \quad , \quad (4)$$ $l=1$ where $_l$: .

III. Preliminaries: Impacts of Feature Length and AoI on Inference Error

In this section, we adopt an information-theoretic approach that was developed recently in [7] to show the impact of feature length l and AoI δ on the inference error $\text{err}_{\text{inference}}(\delta,l)$.

A. Information-Theoretic Metrics for Training and Inference Errors

Training error $\text{err}_{\text{training}}(\delta,l)$ is expressed as a function of δ and l, given by

$$\text{err}_{\text{training}} \quad , \quad (5)$$

where ϕ_l a trained neural network used in (1) and $P_{Y_0,X_{-l}}$ inis the joint distribution of the target Y_0 and the feature X_δ

the training dataset. The training error $\text{err}_{\text{training}}(\delta,l)$ is lower-bounded by

$$H_L \quad , \quad (6)$$

where is the set of all functions that map from $\mathbb{Z}^+ \times \mathbb{V}^l$ to $\mathbb{A}$. Because the trained neural network ϕ_l in (5) satisfies $\phi_l \in$, $H_L(Y_0|X_{-l}) \le \text{err}_{\text{training}}(\delta,l)$. The lower bound in (6) has an informationtheoretical interpretation [7], [41], [42], [43]: It is a generalized conditional entropy of a random variableassociated to the loss function L. For notational simplicity, we Y_0 given X_{-l}

call $H_L(Y|X)$ an L-conditional entropy of a random variable Y given X. The *L-entropy* of a random variable Y is defined as [41], [42]

$$H_L(Y) \quad . \quad (7)$$

The optimal solutions to (7) may not be unique. Let a_{PY} denote an optimal solution to (7), which is called a *Bayes action* [41]. Similarly, the L-conditional entropy of Y given $X = x$ is defined as [6], [7], [41], [42]

$$H_L(Y|X) \quad [L(Y,a)] \quad (8)$$

and the L-conditional entropy of Y given X is given by [6], [7], [41], [42]

$$H_L(Y|X) \quad x). \quad (9)$$

The inference error $\text{err}_{\text{inference}}(\delta,l)$ can be approximated as the following L-conditional cross entropy

$$H_L \quad x\in\mathbb{X}\ _{t-\delta}(x)\,\mathbb{E}_{Y\sim P_{Y_t|X_{t-\delta}=x}} \quad Y,a_{P_{Y_0|X_{-}}} \quad _{\delta=x} \quad , \quad (10)$$

defined aswhere the L-conditional cross entropy $H_L(P_{Y|X};P_{Y_{-}|X_{-}}|P_X)$ is

[7]

$$H_L P_{Y|X};P_{Y_{-}|X_{-}}|P_X$$

$$= \quad _{Y|X=x} \quad Y,a_{P_{Y_{-}|X_{-}=x}}. \quad (11)$$

If training algorithm considers sets of large and wide neural networks such that $a_{P_{l\,x}}$ and $\phi_l(\delta,x)$ for all $_{Y_0|X_{-\delta}=}$ $\delta \in \mathbb{Z}_+$ and $x \in \mathbb{X}^l$ are close to each other, then the difference between the inference error $\text{err}_{\text{inference}}(\delta,l)$ and thesmall$L$-[7]conditional cross entropy $H_L(P_{Y_t|X_{t-}}$-L_δ;-conditional cross$P_{Y_0|X_{-l\,\delta}}|P_{X_{t-\delta}})$ is

. Compared to $\text{err}_{\text{inference}}(\delta,l)$, the entropy $H_L(P_{Y_t|X_{t-\delta}};P_{Y_0|X_{-l\ \delta}}|P_{X_{t-\delta}})$ are mathematically more convenient to analyze, as we will see next.

B. Information-Theoretic Monotonicity Analysis

The following lemma interprets the monotonicity of the L-conditional entropy $H_L(Y|X^l)$ and the L-conditional cross entropy $H^L(P_{Y_t|X_{t-\delta}};P_{Y|X^{l-\delta}}|P_{X_{t-\delta}})$ with respect to the feature length l.

Lemma 1: The following assertions are true:

(a) Given l, i.e., for all $1\delta \geq 0$, $H_L(Y|X^{l-\delta})$ is a non-increasing function of

$$H_L \quad . \tag{12}$$

(b) Given $\beta \geq 0$, if for all $l = 1,2,\ldots$, and $x \in \mathbb{V}^l$

$$\sum_{X_t} P_{Y^t|X_t^{l-\delta}=x}(y) - P_{Y|X^{l-\delta}=x} \Big| \leq 2\beta, \tag{13}$$

then for all $1 \leq l_1 \leq l_2$

$$H^L \quad P_t \quad \delta; \quad 0 \quad 2\delta \Big| \quad t2 \quad Y|X - \quad Y \quad |X - \quad X - \delta \leq H_L \quad P \quad l_1; \quad 0 \quad 1\delta \Big| \quad t1 + O(\beta). \tag{14}$$

$$Y_t|X_{t-\delta} \quad Y|X - \quad X - \delta$$

Proof: Lemma 1 can be proven by using the data processing inequality for L-conditional entropy [43, Lemma 12.1] and a local information geometric analysis. See Appendix B of the supplementary material for the details. ■

Lemma 1(a) demonstrates that for a given AoI value δ, the L-conditional entropy $H_L(Y|X^{l-\delta})$ decreases as the feature length l increases. This is due to the fact that a longer feature provides more information, consequently leading to a lower L-conditional entropy. Additionally, as indicated in Lemma 1(b), when the conditional distributions in training and inference data are close to each other (i.e., when β in (13) is close to 0), the L-conditional cross entropy $H_L(P_{Y_t|X_{t-\delta}};P_{Y|X^{l-\delta}}|P_{X_{t-\delta}})$ is close to a non-increasing function of the feature length l. This information-theoretic analysis clarifies the experimental results depicted in Fig. 2(a) and Fig. 3(a), where the inference error diminishes with the increasing feature length.

The monotonicity of the L-conditional cross entropy $H_L(P_{Y_t|X_{t-\delta}};P_{Y|X^{l-\delta}}|P_{X_{t-\delta}})$ with respect to the AoI δ are explained in [7, Th. 3] and in [35]. This result is restated in Lemma 2 below for the sake of completeness.

Definition 1 (-Markov Chain [7], [35]): Given ≥ 0, a sequence of three random variables Y,X, and Z is said to be an *-Markov chain*, denoted as $Y \leftrightarrow X \leftrightarrow Z$, if

$$I_{\log}(Y;Z|X) \quad P_{Y|X,Z}||P_{Y|X}$$

where

$$D_{\log}(P_Y||Q_Y) = \quad (y) \quad Y \quad \text{(16)} \quad Q_Y(y) \quad y \in \mathcal{Y}$$

is KL-divergence and $I_{\log}(Y;Z|X)$ is Shannon conditional mutual information.

-Markov chain for all $\mu,\nu \geq \leftrightarrow$ 0 and ^{l} (13) $\leftrightarrow$ $X_{t-\mu-\nu}^l$ is an *Lemma 2 [7], [35]:* If $Y_t X_{t-\mu}$ holds, then for all $0 \leq \delta_1 \leq \delta_2$

$$H_L \; P_{Y_t|X_t} \quad \leq H_L \quad P_{Y_t|X_t}$$

Lemma 2 implies that the monotonic behavior of $H_L(P_{Y_t|X_{t-\delta}};P_{Y|X^{l-\delta}}|P_{X_{t-\delta}})$ with respect to AoI δ is characterized by two key parameters: in the - Markov chain model and the parameter β. When is small, the sequence of target and feature random variables approximates a Markov chain.

Consequently, decreasing with respect $H^L(P_{Y_t}$ to $|X_t$ -$_\delta$AoI$;P_{Y|^\delta X^{l-\delta}}$ provided $|P_{X_{t-\delta}})$ becomes non-that β is close to 0. Conversely, if is significantly large, then $H_L(P_{Y_t|X_{t-\delta}};P_{Y|X^{l-\delta}}|P_{X_{t-\delta}})$ can be far from a monotonic function of δ. This findings provide an explanation for the

patterns observed in the experimental results shown in Figs. 2(b) to 3(b). Shannon's interpretation of Markov sources in his seminal work [44] indicates that as the sequence length l grows larger, the tuple $(Y_t, X_{t-\mu}^{l}, X_{t-\mu-\nu}^{l})$ tends to resemble a Markov chain more closely. Hence, according to Lemma 2, the inference error approaches to a non-decreasing function of AoI δ as feature length l increases. As illustrated in Figs. 2(b)-3(b), the inference error converges to a nondecreasing function of AoI δ as feature length l increases.

IV. Learning and Communications Co-Design: Single Source Case

Let $d(t)$ denote the feature length of the most recently received feature in time slot t. The time-averaged expected inference error under policy $\pi = ((S_1,b_1,l_1),(S_2,b_2,l_2),\ldots)$ is expressed as

[redacted] $\mathrm{err}_{\text{inference}}$ [redacted], (18)

where $\bar{p}_\pi$ is denoted as the time-averaged inference error, and $\mathrm{err}_{\text{inference}}((t),d(t))$ is the expected inference error at time t corresponding to the system state $((t),d(t))$. In this section, we solve two problems. The first one is to find an optimal policy that minimizes the time-averaged expected inference error among all the causal policies in $_{\text{inv}}$ that consider timeinvariant feature length. Another problem is to find an optimal policy that minimizes the time-averaged expected inference error among all the causal policies in .

A. Time-Invariant Feature Length

We first find an optimal policy that minimizes the timeaveraged inference error among all causal policies with timeinvariant feature length in $_{\text{inv}}$ defined in (4):

[redacted] $\mathrm{err}_{\text{inference}}$ [redacted]

where $\bar{p}_{\text{inv}}$ is the optimum value of (19). The problem (19) is an infinite time-horizon average-cost semi-Markov decision process (SMDP). Such problems are often challenging to solve analytically or with closed-form solutions. The perslot cost function $\mathrm{err}_{\text{inference}}((t),d(t))$ in (19) depends on two variables: the AoI (t) and the feature length $d(t)$. Prior studies [9], [11], [12], [13], [14], [18], [19], [21], [45] have considered linear and non-linear monotonic AoI functions. Due to the fact that (i) the cost function in (19) depends on two variables and (ii) is not necessarily monotonic with respect to AoI, finding an optimal solution is challenging and the existing scheduling policies cannot be directly applied to solve (19). Therefore, it is necessary to develop a new scheduling policy that can address the complexities of (19).

Surprisingly, we get a closed-form solution of (19). To present the solution, we define a function $\gamma_l(\delta,d)$ as

[redacted] τ $1\ l$ = [redacted]

$-$ $\mathrm{err}_{\text{inference}}$.

$\tau\in\{1,2,\ldots\}$ τ $j=0$

(20)

Theorem 1: If $T_i(l)$'s are i.i.d. with a finite mean $\mathbb{E}[T_i(l)]$ for each $l = 1,2,\ldots,B$, then there exists an optimal solution [redacted] $_{\text{inv}}$ to (19) that satisfies:

[redacted] (a) The optimal feature position in π_* is time-invariant, i.e., $b_{*1} b_*$. The optimal feature length l_* and the optimal feature position b_* in π_* are given by

[redacted] [redacted] $\beta_{b,l}$, (21)

$1\le l\le B, 0\le b\le B-l$

where $\beta_{b,l}$ is the unique root of equation

[redacted] $\mathrm{err}_{\text{inference}}$ [redacted]

[redacted], (22)

D_i [redacted] b, the sequence $(S_1(\beta_{b,l}),S_2(\beta_{b,l}),\ldots)$ is determined by

S_i [redacted],

(23)

and the function $\gamma_l(\cdot)$ is defined in (20).

(b) The optimal scheduling time S^*_{i+1} in π_* is determined by

S [redacted],

(24)

where [redacted] b_* is the AoI at time t. The optimal objective value $\bar{p}_{\text{inv}}$ of (19) is

[redacted] $\beta_{b,l}$. (25)

$B-l$

We prove Theorem 1 in two steps: (i) We find B policies, each of which is optimal among the set of policies $_l$ where $l = 1,2,\ldots,B$. After that (ii) we select the policy that results in the minimum average inference error among the B policies. See Appendix C of the supplementary material for details.

Theorem 1 implies that the optimal scheduling policy has a nice structure. According to Theorem 1(a), the feature position b_{*i} is constant for all i-th features, i.e., b_* [redacted].

The optimal feature length l_* and the optimal feature position b_* are pre-computed by solving (21) and then used in real-time. The parameter $\beta_{b,l}$ in (21) is the unique root of (22), which is solved by using low-complexity algorithms, e.g., bisection search, newtons method, and fixed point iteration [12]. Theorem 1(b) implies that the optimal schedul-

is transmitted in time-sloting time S_{i*+1} follows a threshold policy. Specifically, a feature$t$ if the following two conditions are

satisfied: (i) The channel is idle in time-slot t and (ii) the value [redacted] exceeds the optimal objective value p^-_{inv} of (19). The optimal objective value p^-_{inv} is obtained from (25). Our threshold-based scheduling policy has a significant distinction from previous threshold-based policies studied in the literature, such as [11], [12], [13], [21]. In these prior works, the threshold function used to determine the scheduling time is based solely on the AoI value and is non-decreasing with respect to AoI. However, in our proposed strategy, (i) the threshold function $\gamma_l(\cdot)$ depends on both the AoI value and the feature length and (ii) the threshold function $\gamma_l(\cdot)$ can be non-monotonic with respect to AoI.

1) Monotonic AoI Cost function: Consider a special case where the inference error $\text{err}_{\text{inference}}(\delta,l)$ is a non-decreasing function of δ for every feature length l. A simplified solution can be derived for this specific case of (19). In this scenario, the optimal feature position is $b^* = 0$, and the threshold function $\gamma_l(\cdot)$ defined in (20) becomes:

$$\gamma_l(\delta,d) \text{ [redacted] } \text{err}_{\text{inference}} \text{ [redacted]}. \tag{26}$$

In this special case of monotonic AoI cost function, (24) can be rewritten as a threshold policy of the AoI (t) in the form of $(t) \geq w(l_*,p^-_{\text{inv}})$, where $w(l,\beta)$ is defined as:

$$w \text{ [redacted] } \text{err}_{\text{inference}} \text{ [redacted]}$$

However, when $\text{err}_{\text{inference}}(\delta,l)$ is not monotonic with respect to AoI δ, (24) cannot be reformulated as a threshold policy of the AoI (t). This is a key difference with earlier studies [11], [13], [14].

2) Connection With Restart-in-State Problem: Consider another special case in which all features take 1 time-slot for transmission. For this special case, the threshold function $\gamma_l(\cdot)$ defined in (20) becomes

$$\text{[redacted]} \quad \text{err}_{\text{inference}} \text{ [redacted]} \quad =0$$

This special case of (19) is a restart-in-state problem [46, Ch. 2.6.4]. This is because whenever a feature with the optimal feature length l_* and from the optimal feature position b_* is transmitted, AoI value restarts from $b^* + 1$ in the next time slot. For this restart-in-state problem, the optimal sending time follows a threshold policy [46, Ch. 2.6.4]. Specifically, a feature is transmitted if

$$h \text{ [redacted]}, \tag{29}$$

where the relative value function $h(\delta,l_*)$ of the restart-in-state problem is given by

$$h \text{ [redacted] } \text{err}_{\text{inference}} \text{ [redacted]} \tag{30}$$

By using (30), we can show that (29) is equivalent to

$$\text{[redacted]}. \tag{31}$$

where the function $\gamma_l(\delta,d)$ is defined in (28). This connection between the restart-in-state problem and AoI minimization was unknown before. The original problem considers more general $T_i(l)$, which can be considered as a restart-in-random state problem. This is because whenever i-th feature with optimal feature length l_* and from optimal feature position b_* is transmitted, AoI restarts from a random value $b^*+T_i(l_*)$ after $T_i(l_*)$ time slots.

B. Time-Variant Feature Length

Now, we find an optimal scheduling policy that minimizes time-averaged inference error among all causal policies in :

$$\text{[redacted]} \quad \text{err}_{\text{inference}} \text{ [redacted]}$$

where $\text{err}_{\text{inference}}$ [redacted] is the inference error at time slot t and p^-_{opt} is the optimum value of (32). Because $_{\text{inv}} \subset$,

$$p^-_{opt} \leq {}^-p_{\text{inv}}, \tag{33}$$

where p^-_{inv} is the optimum value of (19). Like (19), problem (32) can also be expressed as an infinite time-horizon average-cost SMDP. Note that (32) is more complex SMDP than (19) because the feature length in (32) is allowed to vary over time.

The optimal policy can be determined by using a dynamic programming method associated with the average cost

SMDP [15], [47]. There exists a function $h(\cdot)$ such that for all $\delta \in Z_+$ and $0 \le d \le B$, the optimal objective value p^-_{opt} of (32) satisfies the following Bellman equation: $h(\delta,d)$

[redacted] $\mathrm{err}_{\mathrm{inference}}$

$$\sum_{b\in:0\le b\le B-l} \quad k=0$$

$$+ \mathsf{E}[h(T_1(l)+b,l)]. \tag{34}$$

Let $(Z_*(\delta,d),l_*(\delta,d),b_*(\delta,d))$ be the optimal solution to the Bellman equation (34). There exists an optimal solution [redacted] to (32), determined by

$$l \text{ [redacted]}_{+1} \quad i \; i \; i \; , \; i \; , \tag{35}$$

b [redacted], (36)

$$S \text{ [redacted]}_{+1\, i\, i\, i\, i\, i\, i\, , \, i \, ,}$$

(37) where Z[redacted] is the optimal waiting time for sending the $(i+1)$-th feature after the i-th feature is delivered. To get the optimal policy π_*, we need to solve (34). Solving (34) is complex as it requires joint optimization of three variables. Moreover, an optimal solution obtained by the dynamic programming method provides no insight. We are able to simplify (34) in Theorem 2 by analyzing the structure of the optimal solution.

Theorem 2: The following assertions are true:

(a) If $T_i(l)$'s are i.i.d. with a finite mean $\mathsf{E}[T_i(l)]$ for each $l = 1,2,\ldots,B$, then there exists a function $h(\cdot)$ such that for all $\delta \in Z_+$ and $0 \le d \le B$, the optimal objective value p^-_{opt} of (32) satisfies the following Bellman equation:

[redacted] $h\mathrm{err}_{\mathrm{inference}}(\delta + k,d) - p^-_{opt}$]

[redacted], (38)

where $h(\cdot)$ is called the relative value function and the function $Z_l(\delta,d)$ is given by

Z_l [redacted], (39)

and the function $\gamma_l(\delta,d)$ is defined in (20).

(b) In addition, there exists an optimal solution $\pi^* =$ [redacted] to (32) that is determined by

[redacted] l_{i*} [redacted] $1\le l\le B$ $k=0$]

[redacted] $\mathrm{err}_{\mathrm{inference}}$ [redacted]

$$+ \min_{0\le b\in\le B-l} \mathsf{E}[h(T_1(l)+b,l), \tag{40}$$

[redacted]

b

$$+1$$

S [redacted], (42)

where [redacted] is the AoI at time t and $D_i = S_i^* + T_i(l_{i*})$ is the i-th feature delivery time.

Theorem 2(a) simplifies the Bellman equation (34) to (38). Unlike (34), which involves joint optimization of three variables, (38) is an integer optimization problem. This simplification is possible because, for a given feature length l, the original equation (34) can be separated into two separated optimization problems. The first problem involves finding the optimal stopping time, denoted by $Z_l(\delta,d)$ defined in (39), and the second problem is to determine the feature position b that minimizes $\mathsf{E}[h(T_1(l)+b,l)]$. By breaking down the original equation in this way, we can solve the problem more efficiently. Detailed proof of Theorem 2 can be found in Appendix D of the supplementary material.

Furthermore, Theorem 2(a) provides additional insights into the solution of (34). Theorem 2(a) implies that the optimal stopping time $Z_*(\delta,d)$ in (34) follows a threshold policy. Specifically, if $l_*(\delta,d) = l$, then $Z_*(\delta,d)$ equals $Z_l(\delta,d)$, which is defined in (39). Here, $Z_l(\delta,d)$ is the minimum positive integer value τ for which $\gamma_l(\delta+\tau,d)$ defined in (20) exceeds the optimal objective value p^-_{opt}.

Theorem 2(b) provides an optimal solution $\pi^* \in$ to (32). According to Theorem 2(b), by using precomputed p^-_{opt} and the relative value function $h(\cdot)$, we can obtain the optimal feature

After obtaininglength l_{i*+1} froml_{i*}(40)+1, the optimal feature

positionusing an exhaustive search algorithm.b_{*i+1} can

provided inbe determined from(42) follows a threshold

policy. Specifically, the(41). The optimal scheduling time

S_{i*+1}

$(i+1)$-th feature is transmitted in time-slot t if two conditions are satisfied: (i) the previous feature is delivered by time t, and

(ii) the [redacted] functionexceeds the optimal objective [redacted]+

value p^-_{opt} of (32).

1) Policy Iteration Algorithm for Computing p^-_{opt} and $h(\cdot)$: To effectively implement the optimal solution $\pi^* \in$ for (32), as outlined in Theorem 2, it is necessary to precompute the optimal objective value p^-_{opt} and the relative value function $h(\cdot)$ that satisfies the Bellman equation (38). The computation of p^-_{opt} and $h(\cdot)$ can be achieved by employing policy iteration algorithm or relative value iteration algorithm for SMDPs, as detailed in [15, Sec. 11.4.4]. To apply the relative value iteration algorithm, we need to transform the SMDP into an equivalent MDP. However, this transformation process can be challenging to execute. Therefore, in this paper, we opt to utilize the policy iteration algorithm specifically tailored for SMDPs [15, Sec. 11.4.4]. Algorithm 2 provides a policy iteration algorithm for obtaining p^-_{opt} and $h(\cdot)$, which is composed of two steps: (i) *policy evaluation* and (ii) *policy improvement*.

Policy Evaluation: Let h_π and p^-_π be the relative value function and the average inference error under policy π. Let $l_\pi(\delta,d)$, $b_\pi(\delta,d)$, and $Z_\pi(\delta,d)$ represent feature length, feature position, and waiting time for sending the $(i+1)$-th feature under policy π when $(D_i) = \delta$ and $d(D_i) = d$. Given $l_\pi(\delta,d)$, $b_\pi(\delta,d)$, and $Z_\pi(\delta,d)$ for all (δ,d), we can Algorithm 1 Policy Evaluation Algorithm

1: Input: $Z_\pi(\delta,d)$, $l_\pi(\delta,d)$, and $b_\pi(\delta,d)$ for all (δ,d).
2: Initialize $h_\pi(\delta,d)$ arbitrarily for all (δ,d), except for one fixed state (δ,d) with $h_\pi(\delta,d) = 0$.
3: Initialize a small positive number α_1 as a threshold.
4: repeat
5: $\theta_1 \leftarrow 0$.
6: Determine p^-_π using (43).
7: for [redacted] each state (δ,d) do
[redacted]
8: $\pi(\delta,$ $(l_\pi(\delta,d))]$.
9: h (δ,d) E % – $\text{err}_{\text{inference}}(\delta + k,d) - p^-_\pi)]$ + $[h_\pi(T_1(l_\pi(\delta,d)) + \pi(\delta,d),l_\pi(\delta,d))]$.
10: $\theta_1 \leftarrow$ max.
11: end for
12: $h_\pi \leftarrow h_\pi$.
13: until $\theta_1 \le \alpha_1$. 14:
return p^-_π and $h_\pi(\cdot)$.

evaluate the relative value function $h_\pi(\cdot)$ and the average inference error p^-_π using Algorithm 1. The relative value function $h_\pi(\delta,d)$ represents relative value associated with a reference state. We can set (δ,d) as a reference state with $h_\pi(\delta,d) = 0$. By using h[redacted] 0, the average inference error p^-_π is determined by

[redacted]$\text{err}_{\text{inference}}$[redacted], (43)

where [redacted] $T_1(l_\pi(\delta,d))]$. We then use an iterative procedure within Algorithm 1 to determine the relative value function $h_\pi(\cdot)$.

Policy Improvement: After obtaining h_π and $p^-\pi$ from Algorithm 1, we apply Theorem 2 to derive an improved policy π in Algorithm 2. Feature length $l_\pi(\delta,d)$, feature position $b_\pi(\delta,d)$, and waiting time $Z_\pi(\delta,d)$ under policy π is determined by

[redacted] $Z(\delta,d)$ $T(l)$ 1 [redacted] l [redacted] $\text{err}_{\text{inference}}(\delta + k,d)$

$=0$

$- \mathrm{E}[Z_l(\delta,d) + T_1(l)]p^-\pi$

$+ \min_{0\le b\le B-l} \mathrm{E}[h_\pi(T_1(l) + b,l)]$, (44)

b [redacted] b,l π

(45)

Z [redacted]

Instead of a joint optimization problem (34), Algorithm 2 utilizes separated optimization problems (44)–(46) based on Theorem 2. If the improved policy π is equal to the old policy π, then the policy iteration algorithm converges.

Algorithm 2 Policy Iteration Algorithm

1: Initialize $Z_\pi(\delta,d)$, $l_\pi(\delta,d)$, and $b_\pi(\delta,d)$ for all (δ,d).
2: Initialize a small positive number α_2 as threshold.
3: repeat
4: $\theta_2 \leftarrow 0$.
5: Obtain $h_\pi(\cdot)$ and $p^-\pi$ from Algorithm 1.
6: for all (δ,d) do
7: Get l [redacted] using (44)-(46).
8: [redacted]
9: [redacted] $|+|Z$ [redacted].
$l_\pi(\delta,d) \leftarrow l$ [redacted] $b_\pi(\delta,d) \leftarrow b$ [redacted]
$Z_\pi(\delta,d) \leftarrow Z$ [redacted] end for
until $\theta_2 \le \alpha_2$. return $p^-opt \leftarrow {}^-p_\pi$ and $h \leftarrow h_\pi$.

[15, Th. 11.4.6] establishes the finite convergence of the policy iteration algorithm of an average cost SMDP.

Now, we discuss the time-complexity of Algorithms 1-2. To manage the infinite set of AoI values in practice, we introduce an upper bound denoted as δ_{bound}. Whenever δ exceeds δ_{bound}, we set $h_\pi(\delta,d) = h_\pi(\delta_{\text{bound}},d)$ for all d. Hence, each iteration of our policy evaluation step requires one pass through the approximated state space $\{1,2,\ldots,\delta_{\text{bound}}\} \times \{1,2,\ldots,B\}$.

Therefore, the time complexity of each iteration is $O(\delta_{\text{bound}}B)$, assuming that the required expected values are precomputed. Considering the bounded set $\{0,1,\ldots,\delta_{\text{bound}}\}$ instead of $\mathbb{Z}+$, the time complexities of (44), (45), and (46) are $O(B^2)$, $O(B)$, and $O(\delta_{\text{bound}})$, respectively, provided that the expected values in (44)–(46) are precomputed. The overall complexity of (44)–(46) is $O(\max\{B^2,B,\delta_{\text{bound}}\})$, which is more efficient than the joint optimization problem (34). The latter has a time complexity of $O(\delta_{\text{bound}}B^2)$. In each iteration of the policy improvement step, the optimization problems (44)–(46) are solved for all state (δ,d) such that $\delta = 1,2,\ldots,\delta_{\text{bound}}$ and $d = 1,2,\ldots,B$. Hence, the total complexity of each iteration of the policy improvement step is $O(\max\{B^3\delta_{\text{bound}}, B\delta_{\text{bound}}^2\})$.

V. Learning and Communications Co-Design: Multiple Source Case

A. System Model

Consider a remote inference system consisting of $M \ge 1$ source-predictor pairs connected through $N \ge 1$ shared communication channels, as illustrated in Fig. 4. Each source j has a buffer that stores B_j most recent signal observations at each time slot t. At time slot t, a centralized scheduler determines whether to send a feature from source j with feature length $l_j(t)$ and feature position $b_j(t)$. We denote $l_j(t) = 0$ if the scheduler decides not to send a feature from source j at time t. If a feature from source j is sent, we assume it will be delivered to the j-th neural predictor in the next time slot using $l_j(t)$ channel resources. The transmission model of the multiple source system is significantly different from that of

Fig. 4. A multiple source-predictor pairs and multiple channel remote inference system.

the single source model discussed in Section II-C. In the latter case, only one channel was considered, while N communication channels are available in the former. The channels could be from multiple frequencies and/or time resources. For example, if the clock rate in the multiple access control (MAC) layer is faster than that of the application layer, then one applicationlayer time-slot could comprise multiple MAC-layer time-slots. A feature can utilize multiple channels (i.e., frequency or time resources) for transmission during a single time slot. However, the channel resource is limited, so the system must satisfy

$$\sum_{j=1}^{M} [redacted]. \quad (47)$$

The system begins operating at time $t = 0$. Let $S_{j,i}$ denote the sending time of the i-th feature from the j-th source. Since we assume that a feature takes one time-slot to transmit, the corresponding neural predictor receives the i-th feature from the j-th source at time $S_{j,i}+1$. The AoI of the source j at time slot t is defined as

$$[redacted], \text{ if } S_{j,i} < t \le S_{j,i+1}. \quad (48)$$

We denote $d_j(t)$ as the feature length of the most recent received feature from j-th source by time t, given by

$$d_j[redacted], \text{ if } S_{j,i} < t \le S_{j,i+1}. \quad (49)$$

B. Scheduling Policy

At time slot t, a centralized scheduler determines the value of the feature length $l_j(t)$ and the feature position $b_j(t)$ for every j-th source. A scheduling policy is denoted by

πLet=(πdenote the set of all the causal scheduling policies$_j)_{j=1}^{M}$, where $\pi_j = ((l_j(1),b_j(1)),(l_j(2),b_j(2)),\ldots)$.

that determine $l_j(t)$ and $b_j(t)$ based on the current and the historical information available at the transmitter such that $0 \le l_j(t) + b_j(t) \le B_j$.

C. Problem Formulation

Our goal is to minimize the time-averaged sum of the inference errors of the M sources, which is formulated as

$$(50) \quad [redacted] \quad ,$$

$$j=1 \qquad t=0$$

$$\text{s.}[redacted] \; N,\ t = 0,1,2,\ldots, \quad (51)$$

$$j=1$$

where $p_j(j(t),d_j(t))$ is the inference error of source j at time slot t.

The problem (50)–(51) can be cast into an infinitehorizon average cost restless multi-armed multi-action bandit problem [17], [39] by viewing each source j as an arm, where

a scheduler needs to decide multiple actionsevery time t by observing state $(j(t),d_j(t)).(l_j(t),b_j(t))_j^M{}_{=1}$ at

Finding an optimal solution to the RMAB problem is PSPACE hard [16]. Whittle, in his seminal work [17], proposed a heuristic policy for RMAB problem with binary action. In [39], a modified Whittle index policy is proposed for the multi-action RMAB problems. Whittle index policy is known to be asymptotically optimal [48], but the policy needs to satisfy a complicated indexability condition. Proving indexability is challenging for our multi-action RMAB problem because we allow (i) general penalty function $p_j(\delta,l)$ that is not necessarily monotonic with respect to AoI δ and (ii) time-variant feature length. To this end, we propose a low-complexity algorithm that does not need to satisfy any indexability condition.

D. Lagrangian Optimization of a Relaxed Problem

Similar to Whittle's approach [17], we utilize a Lagrange relaxation of the problem (50)–(51). We first relax the per time-slot channel constraint (51) as the following time-average expected channel constraint

$$[redacted] \quad .$$

$$(52)$$

$$j$$

The relaxed constraint (52) only needs to be satisfied on average, whereas (51) is required to hold at every time-slot. By this, the original problem (50)–(51) becomes

[redacted], (53) s.

[redacted] . (54)

The relaxed problem (53)–(54) is of interest as the optimal solution of the problem provides a lower bound to the original problem (50)–(51).

1) Lagrangian Dual Decomposition of (53)–(54): To solve (53)–(54), we utilize a Lagrangian dual decomposition method [17], [49]. At first, we apply Lagrangian multiplier $\lambda \geq 0$ to the time-average channel constraint (54) and get the following [redacted] Lagrangian dual function $q - \lambda N$.

The problem (55) can be decomposed into M sub-problems. The sub-problem associated with the j-th source is defined as:

[redacted], (56)

where $_j$ is the set of all causal scheduling policies π_j. The sub-problem (56) is an infinite horizon average cost MDP, where a scheduler decides action $(l_j(t),b_j(t))$ by observing state $(_j(t),d_j(t))$. The Lagrange multiplier λ in (56) can be interpreted as a transmission cost: whenever $l_j(t) = l$, the source j has to pay cost of λl for using l channel resources.

The optimal solution to (56) can be obtained by solving the following Bellman equation:

$$h_{j,\lambda}[redacted] Q_{j,\lambda}((\delta,d),(l,b)), \quad (57)$$

$0 \leq l+b \leq B_j$

where $h_{j,\lambda}(\cdot)$ represents the relative value function of the MDP (56), and the function $Q_{j,\lambda}(\cdot,\cdot)$ is defined as follows

$$Q_{j,\lambda}((\delta,d),(l,b)) := [redacted] \quad (58)$$

$= \delta,\ \delta\ ,\ $ if $l\ 0, p_j(\delta,d)p_j(\lambda)$

$h_{j,\lambda}(b\ \ 1,l)\ \ \lambda l$, otherwise.

The relative value function $h_{j,\lambda}(\cdot)$ can be computed using the relative value iteration algorithm [15], [47].

Let $\pi_{j^*,\lambda} = ((l_{j^*,\lambda}(1),b_{*j,\lambda}(1)),(l_{j^*,\lambda}(2),b_{*j,\lambda}(2)),\ldots)$ be an optimal solution to (56), which is derived by using (57) and (58). The optimal feature length $l_{j^*,\lambda}(t)$ is determined by

$$l_{j^*,\lambda}(t) h_{j,\lambda} h_{j,\lambda} [redacted] \quad (59)$$

B_j

$\times$ [redacted] l,

where the function $\hat{b}_{j,\lambda}(l)$ is given by

$$\hat{b}_{j,\lambda}(l) [redacted] 1,l), \quad (60)$$

The optimal feature position in [redacted] is b

[redacted] $_{,\lambda}$. (61)

2) Lagrange Dual Problem: Next, we determine the optimal dual cost λ_* that solves the following Lagrange dual problem:

$$\max_{\lambda \geq 0} q(\lambda), \quad (62)$$

where $q(\lambda)$ is the Lagrangian dual function defined in (55). To get λ_*, we apply the stochastic sub-gradient ascent method [49], which iteratively updates $\lambda(k)$ as follows [redacted]

M

$\lambda(k$ [redacted] , (63)

where k is the iteration index, $\beta > 0$ determines the step size $_{-k}$, and $l_{j,\lambda(k)}(k)$ is the feature length of source j at the k-th iteration. Detailed optimization technique is provided in Algorithm 3.

E. Net Gain Maximization Policy

[redacted] After getting optimal dual cost λ_*, we can use policy for the relaxed problem (53)–(54). But it is infeasible to implement the policy for the original problem (50)–(51) because it may violate the scheduling ~~constraint (51). Motivated by Whittle's approach [17], we aim to select actions Algorithm 3 Dual Algorithm to Solve (62)~~

Input: Step size $\beta > 0$ and dual cost $\lambda(1) = 0$.

Initialize $_j(0)$, $d_j(0)$, $l_j(0)$, and $b_j(0)$ for all j.

Initialize a small positive number θ as threshold.

repeat

for each source j do if $l_j(k-1) > 0$ then

[redacted]

$_j(k) \leftarrow 1 + b_j(k-1)$, $d_j(k) \leftarrow l_j(k-1)$.

else

$_j(k) \leftarrow _j(k-1) + 1$, $d_j(k) \leftarrow d_j(k-1)$.

end if

[redacted]

Compute $l_{j,\lambda(k)}(k)$ using (59).

Compute $b_{j,\lambda(k)}(k)$ using (61).

end for

[redacted]

Update $\lambda(k+1)$ using (63). until $|\ | \le \theta$. return

with higher priority, while satisfying the scheduling constraint (51) at every time slot. Towards this end, we introduce "Net Gain", denoted as $\alpha_{j,\lambda}(\delta,d,l)$, to measure the advantage of selecting feature length l, which is given by

$\alpha_{j,\lambda}(\delta,d,l)$

:[redacted],$\hat{b}_{j,\lambda}$[redacted],$\hat{b}_{j,\lambda}$[redacted]

where the function $Q_{j,\lambda}$ is defined in (58) and the function $\hat{b}_{j,\lambda}$ is defined in (60). Substituting (58) into (64), we get

$\alpha_{j,\lambda}(\delta,d,l)$ [redacted]

For a given λ, the net gain $\alpha_{j,\lambda}(\delta,d,l)$ has an economic interpretation. Given the state (δ,d) of source j, the net gain $\alpha_{j,\lambda}(\delta,d,l)$ measures the maximum reduction in the loss by selecting source j with feature length l, as opposed to not selecting source j at all. If $\alpha_{j,\lambda}(\delta,d,l)$ is negative for all $l = 1,2,\ldots,B_j$, then it better not to select source j. If $\alpha_{j,\lambda}(_j(t),d_j(t),l_j) > \alpha_{k,\lambda}(_k(t),d_k(t),l_k)$, then the feature length l_j for source j is prioritized over the feature length l_k for source k. Under the constraint (51), we select feature lengths that maximize "Net Gain":

$$\max_{0\le(tl)_j\in(tZ)\le,\forall Bj} \sum_{j=1}^{M} [redacted], \quad (66)$$

$$\text{s.} \sum_{j=1}^{M} [redacted]. \quad (67)$$

The "Net Gain Maximization" problem (66) with constraint (67) is a bounded Knapsack problem. By using (66)–(67), we propose a new algorithm for the problem (50)–(51) in Algorithm 4.

Algorithm 4 starts from $t = 0$. At time $t = 0$, the algorithm takes the dual variable (transmission cost) λ_* from Algorithm 3 which is run offline before $t = 0$. The "Net Gain" $\alpha_{j,\lambda_*}(\delta,d,l)$ is precomputed for every source j, every feature length l, and every state (δ,d) such that Algorithm 4

Net Gain Maximization Policy

Input: Optimal dual variable λ_* obtained in Algorithm 3.

Compute $\alpha_{j,\lambda_*}(\delta,d,l)$ using (65) for all j,δ,d,l.

for each time $t \ge 0$ do

Update $_j(t)$ and $d_j(t)$ using (48) and (49) for all source j.

Compute $(l_j(t))_{j=1}^{M}$ by solving problem [redacted] (66)–(67).

($b_j(t)$)by using (60).

end for

$\delta \in Z_+$, $l,d \in \{1,2,\ldots,B_j\}$, **where we approximate infinite set of AoI values Z_+ by using an upper bound δ_{bound}. We can set $\alpha_{j,\lambda_*}(\delta,d,l) = \alpha_{j,\lambda_*}(\delta_{bound},d,l)$ if $\delta > \delta_{bound}$.**

From time $t \ge 0$, Algorithm 4 solves the knapsack problem (66)–(67) at every time slot t. The knapsack problem is solved by using a dynamic programming method in $O(MNB)$ time [50], where M is the number of sources, N is the number of channels, and B is the maximum buffer size among all source j. The feature position $b_j(t)$ is obtained from a look up table that stores the value of function $\hat{b}_{j,\lambda_*}(l)$ for all j and l.

Unlike the Whittle index policy [17], our policy proposed in Algorithm 4 does not need to satisfy any indexability condition. There exists some other policies that do not need to satisfy indexability condition [36], [38]. The policies in [36], [38] are developed based on linear programming formulations, our policy does not need to solve any linear programming.

VI. Trace-Driven Evaluations

In this section, we demonstrate the performance of our scheduling policies. The performance evaluation is conducted using an inference error function obtained from

a channel state information (CSI) prediction experiment. In Fig. 2, one can observe the inference error function of a CSI prediction experiment. The discrete-time autocorrelation function of the generated fading channel coefficient is defined as $r(k) = bJ_0(2\pi f_d T_s|k|)$, where $r(k)$ represents the autocorrelation of the CSI signal process with time lag k, b signifies the variance of the process, $J_0(\cdot)$ denotes the zeroth-order Bessel function, T_s is the channel sampling duration, $f_d = \frac{vf_c}{c}$ is the maximum Doppler shift, v stands for the velocity of the source, f_c is the carrier frequency, and c represents the speed of light. In this experiment, we employed a quadratic loss function. Although we utilize the CSI prediction experiment and a quadratic loss function for evaluating the performance of our scheduling policies, we note that our scheduling policies are not limited to any specific experiment, loss function, or predictor.

A. Single Source Scheduling Policies

We evaluate the following four single source scheduling policies.

1. Generate-at-Will, Zero Wait with Feature Length l: In this policy, $S_{i+1} = S_i + T_i(l_i)$, $b_i = 0$, and $l_i = l$ for all i-th feature transmissions.
2. Optimal Policy with Time-invariant Feature Length (TIFL): The policy that we propose in Theorem 1.

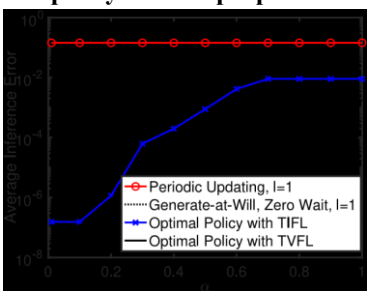


Fig. 5. Single Source Case: Time-averaged inference error vs. the scale parameter α in transmission time $T_i(l) = \alpha l$ for all i.

3. Optimal Policy with Time-variant Feature Length (TVFL): The policy that we propose in Theorem 2.
4. Periodic Updating with Feature Length l: After every time slot T_p, the policy submits features with feature length l and feature position 0 to a First-Come, First-Served communication channel.

We evaluate the performance of the above four single source scheduling policies, where the task to infer the current CSI of a source by observing features. For generating the CSI dataset, we set $b_0 = 1$, $T_s = 1$ms, $v = 15$ m/s, and $f_c = 2$GHz.

Additionally, we add white noise to the feature variable with a variance of 10^{-6}.

In the single source case, we consider that the i-th feature requires $T_i(l) = \alpha l$ time-slots for transmission, where α represents the communication capacity of the channel. For example, if the number of bits used for representing a CSI symbol is n and the bit rate of the channel is ρ, then $\alpha = \frac{\rho}{n}$.

Fig. 5 shows the time-averaged inference error under different policies against the parameter α, where $\alpha > 0$. The plot is constrained to $\alpha = 1$ since values of $\alpha > 1$ is impractical due to the possibility of sending CSI using fewer bits. The buffer size of the source is $B = 10$. Among the four scheduling policies, the "Optimal Policy with TVFL" yields the best performance, while the "Optimal Policy with TIFL" outperforms the other two policies. The findings in Figure 5 demonstrate that when $\alpha \leq 0.1$, the "Optimal Policy with TVFL" can achieve a performance improvement of 10^4 times compared to the "Periodic Updating, $l = 1$" with $T_p = 4$ and "Generate-at-Will, Zero Wait, $l = 1$" policies. This result is not surprising since "Periodic Updating, $l = 1$" and "Generate-atWill, Zero Wait, $l = 1$" do not utilize longer features, despite all features with $l = 1,2,\ldots,10$ taking only 1 time slot when $\alpha \leq 0.1$. When $\alpha > 0.1$, the average inference error of the "Periodic Updating" and "Generate-at-Will, Zero Wait" policies are at least 10 times worse than that of the "Optimal Policy with TVFL." The reasons are as follows: (1) The "Periodic Updating" policy does not transmit a feature even when the channel is available, leading to an inefficient use of resources. In our simulation, this situation is evident as $T_i(1) = 1$ and $T_p = 4$. Again, "Periodic Updating" may transmit features even when the preceding feature has not yet been delivered, resulting in an extended waiting time for the queued feature. This frequently leads to the receiver receiving a feature with a significantly large AoI value, which is not good for accurate inference. (2) Conversely, the "Generate-at-Will, Zero-Wait"

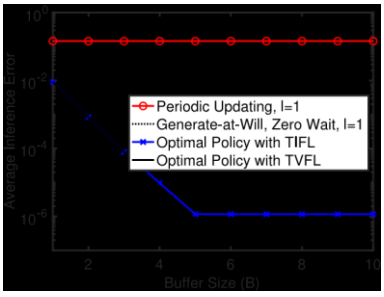


Fig. 6. Single Source Case: Time-averaged inference error vs. the buffer size B.

policy isn't superior because zero-wait is not advantageous, and the feature position $b = 0$ may not be an optimal choice since the inference error is non monotonic with respect to AoI.

The policy "Optimal Policy with TIFL" achieves an average inference error very close to that of the "Optimal Policy with TVFL," but it is simpler to implement. Furthermore, the "Optimal Policy with TIFL" requires only one predictor associated with the optimal time-invariant feature length and does not require switching the predictor.

Fig. 6 plots the time-averaged inference error vs. the buffer size B. In this simulation, $\alpha = 0.2$ is considered. The results show that increasing B can improve the performance of the "Optimal Policy with TVFL" and "Optimal Policy with TIFL" compared to the other policies. As B increases, "Optimal Policy with TVFL" and "Optimal Policy with TIFL" outperform the others. In contrast, the "Periodic Updating" and "Generate-at-Will" policies do not utilize the buffer and their performance remains unchanged with increasing B. Moreover, we can notice that the buffer size $B = 5$ is enough for this experiment as further increase in buffer size does not improve the performance.

B. Multiple Source Scheduling Policies

In this section, we evaluate the time-averaged inference error of the following three multiple source scheduling policies.

1. Maximum Age First (MAF), Generate-at-will, $l = 1$: As the name suggests, this policy selects the sources with maximum AoI value at each time. Specifically, under this policy, $\min\{N,M\}$ sources with maximum AoI are selected. Moreover, the feature length and the feature position of the selected sources are 1 and 0, respectively.
2. Maximum Age First (MAF), Generate-at-will, $l = B$: This policy also selects the sources with maximum AoI values at each time, but with feature length $l = B$. Under this policy, $\min\{\frac{N}{B}, M\}$ sources with maximum AoI are selected, where B is the buffer size of all sources, i.e., $B_j = B$ for all source j. Moreover, the feature position of the selected sources is 0.
3. Proposed Policy: The policy in Algorithm 4.

The performance of three multiple source scheduling policies is illustrated in Fig. 7, where each source sends its observed CSI to the corresponding predictor. In this simulation, three types of sources are considered: (i) type 1 source with a velocity of $v_1 = 15$ m/s and a CSI variance of $b_1 = 0.5$, (ii) type 2 sources with a velocity of $v_2 = 20$ m/s and a CSI

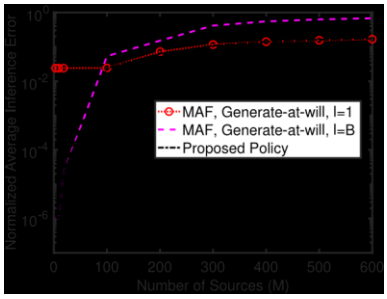


Fig. 7. Multiple Source Case: Time-averaged inference error vs. the number of sources M.

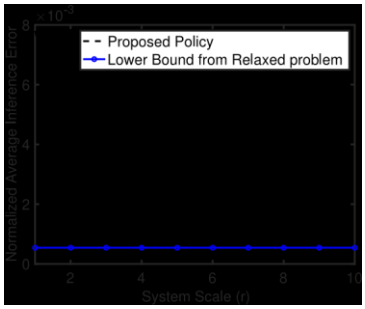


Fig. 8. Multiple Source Case: Time-averaged inference error vs. system scale r, where $M = 3r$ and $N = 10r$.

variance of $b_2 = 0.1$, and (iii) type 3 sources with a velocity of $v_3 = 25$ m/s and a CSI variance of $b_3 = 1$.

Fig. 7 illustrates the normalized average inference error (the total time-averaged inference error divided by the number of sources) plotted against the number of sources M, with $N = 100$ and $B = 10$. We can observe from Fig. 7 that when the number of sources is less, the normalized average inference error of our proposed policy is 10^4 times better than "MAF, Generate-at-will, $l = 1$." However, "MAF, Generate-at-will, $l = B$" is close to the proposed policy. But, When number of sources is more than 400, the normalized average inference error becomes 4 times lower than that of the "MAF, Generateat-will, $l = B$" policy. As the number of sources increases, the normalized average inference error obtained by "MAF, Generate-at-will, $l = 1$" becomes close to the normalized average inference error of the proposed policy.

Fig. 8 compares the time-averaged inference error of the proposed policy and a lower bound from a relaxed problem. The lower bound is achieved by selecting feature length and feature position by using (59) and (61), respectively under

dual cost $\lambda = \lambda_*$ obtained from Algorithm 3. For this evaluation, we have taken step size $10^{-4}/(kr)$ at each iteration k In Algorithm 3. In Fig. 8, we consider $N = 10r$ channels and $M = 3r$ sources, where r represents the system scale. Observing Fig. 8, it becomes evident that our proposed policy converges towards the lower bound as the system scale r increases.

VII. Conclusion

This paper studies a learning and communications co-design framework that jointly determines feature length and transmission scheduling for improving remote inference performance. In single sensor-predictor pair system, we propose two distinct optimal scheduling policies for (i) time-invariant feature length and (ii) time-variant feature length. These two scheduling policies lead to significant performance improvement compared to classical approaches such as periodic updating and zerowait policies. Using the Lagrangian decomposition of a relaxed formulation, we propose a new algorithm for multiple sensorpredictor pairs. Simulation results show that the proposed algorithm is better than the maximum age-first policy.

References

[1] D. C. Nguyen et al., "6G Internet of Things: A comprehensive survey," *IEEE Internet Things J.*, vol. 9, no. 1, pp. 359–383, Jan. 2022.

[2] M. Bojarski et al., "End to end learning for self-driving cars," 2016, *arXiv:1604.07316*.

[3] R. Rahmatizadeh, P. Abolghasemi, L. Bölöni, and S. Levine, "Visionbased multi-task manipulation for inexpensive robots using endto-end learning from demonstration," in *Proc. IEEE ICRA*, 2018, pp. 3758–3765.

[4] C. She et al., "Deep learning for ultra-reliable and low-latency communications in 6G networks," *IEEE Netw.*, vol. 34, no. 5, pp. 219–225, Sep./Oct. 2020.

[5] S. Kaul, R. Yates, and M. Gruteser, "Real-time status: How often should one update?" in *Proc. IEEE INFOCOM*, 2012, pp. 2731–2735.

[6] M. K. Chowdhury Shisher, H. Qin, L. Yang, F. Yan, and Y. Sun, "The age of correlated features in supervised learning based forecasting," in *Proc. IEEE INFOCOM Age Inf. Workshop*, 2021, pp. 1–8.

[7] M. K. Chowdhury Shisher and Y. Sun, "How does data freshness affect real-time supervised learning?" in *Proc. ACM MobiHoc*, 2022, pp. 31–40.

[8] R. D. Yates, "Lazy is timely: Status updates by an energy harvesting source," in *Proc. IEEE ISIT*, 2015, pp. 3008–3012.

[9] Y. Sun, E. Uysal-Biyikoglu, R. D. Yates, C. E. Koksal, and N. B. Shroff, "Update or wait: How to keep your data fresh," *IEEE Trans. Inf. Theory*, vol. 63, no. 11, pp. 7492–7508, Nov. 2017.

[10] V. Tripathi, L. Ballotta, L. Carlone, and E. Modiano, "Computation and communication co-design for real-time monitoring and control in multiagent systems," in *Proc. IEEE WiOpt*, 2021, pp. 65–72.

[11] Y. Sun and B. Cyr, "Sampling for data freshness optimization: Nonlinear age functions," *J. Commun. Netw.*, vol. 21, no. 3, pp. 204–219, Jun. 2019.

[12] T. Z. Ornee and Y. Sun, "Sampling and remote estimation for the Ornstein-Uhlenbeck process through queues: Age of information and beyond," *IEEE/ACM Trans. Netw.*, vol. 29, no. 5, pp. 1962–1975, Oct. 2021.

[13] M. Klügel, M. H. Mamduhi, S. Hirche, and W. Kellerer, "AoI-penalty minimization for networked control systems with packet loss," in *Proc. IEEE INFOCOM Age Inf. Workshop*, 2019, pp. 189–196.

[14] V. Tripathi and E. Modiano, "A Whittle index approach to minimizing functions of age of information," in *Proc. IEEE Allerton*, 2019, pp. 1160–1167.

[15] M. L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. New York, NY, USA: Wiley, 2014.

[16] C. H. Papadimitriou and J. N. Tsitsiklis, "The complexity of optimal queueing network control," in *Proc. IEEE 9th Annu. Conf. Struct. Complexity Theory*, 1994, pp. 318–322.

[17] P. Whittle, "Restless bandits: Activity allocation in a changing world," *J. Appl. Probabil.*, vol. 25, pp. 287–298, 1988.

[18] I. Kadota, A. Sinha, E. Uysal-Biyikoglu, R. Singh, and E. Modiano, "Scheduling policies for minimizing age of information in broadcast wireless networks," *IEEE/ACM Trans. Netw.*, vol. 26, no. 6, pp. 2637–2650, Dec. 2018.

[19] I. Kadota, A. Sinha, and E. Modiano, "Scheduling algorithms for optimizing age of information in wireless networks with throughput constraints," *IEEE/ACM Trans. Netw.*, vol. 27, no. 4, pp. 1359–1372, Aug. 2019.

[20] T. Soleymani, J. S. Baras, and K. H. Johansson, "Stochastic control with stale information–Part I: Fully observable systems," in *Proc. IEEE CDC*, 2019, pp. 4178–4182.

[21] Y. Sun, Y. Polyanskiy, and E. Uysal, "Sampling of the wiener process for remote estimation over a channel with random delay," *IEEE Trans. Inf. Theory*, vol. 66, no. 2, pp. 1118–1135, Feb. 2020.

[22] R. D. Yates, Y. Sun, D. R. Brown, S. K. Kaul, E. Modiano, and S. Ulukus, "Age of information: An introduction and survey," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 5, pp. 1183–1210, May 2021.

[23] B. Sombabu, A. Mate, D. Manjunath, and S. Moharir, "Whittle index for AoI-aware scheduling," in *Proc. IEEE COMSNETS*, 2020, pp. 630–633.

[24] M. Chen, K. Wu, and L. Song, "A Whittle index approach to minimizing age of multi-packet information in IoT network," *IEEE Access*, vol. 9, pp. 31467–31480, 2021.

[25] Y. Hsu, "Age of information: Whittle index for scheduling stochastic arrivals," in *Proc. IEEE ISIT*, 2018, pp. 2634–2638.

[26] J. Sun, Z. Jiang, B. Krishnamachari, S. Zhou, and Z. Niu, "Closed-form Whittle's index-enabled random access for timely status update," *IEEE Trans. Commun.*, vol. 68, no. 3, pp. 1538–1551, Mar. 2020.

[27] A. M. Bedewy, Y. Sun, S. Kompella, and N. B. Shroff, "Optimal sampling and scheduling for timely status updates in multi-source networks," *IEEE Trans. Inf. Theory*, vol. 67, no. 6, pp. 4019–4034, Jun. 2021.

[28] A. M. Bedewy, Y. Sun, R. Singh, and N. B. Shroff, "Optimizing information freshness using low-power status updates via sleep-wake scheduling," in *Proc. ACM MobiHoc*, 2020, pp. 51–60.

[29] J. Pan, A. M. Bedewy, Y. Sun, and N. B. Shroff, "Minimizing age of information via scheduling over heterogeneous channels," in *Proc. ACM MobiHoc*, 2021, pp. 111–120.

[30] T. Soleymani, S. Hirche, and J. S. Baras, "Optimal self-driven sampling for estimation based on value of information," in *Proc. IEEE WODES*, 2016, pp. 183–188.

[31] Y. Sun and B. Cyr, "Information aging through queues: A mutual information perspective," in *Proc. IEEE SPAWC Workshop*, 2018, pp. 1–5.

[32] G. Chen, S. C. Liew, and Y. Shao, "Uncertainty-of-information scheduling: A restless multiarmed bandit framework," *IEEE Trans. Inf. Theory*, vol. 68, no. 9, pp. 6151–6173, Sep. 2022.

[33] G. Chen and S. C. Liew, "An index policy for minimizing the uncertainty-of-information of Markov sources," *IEEE Trans. Inf. Theory*, early access, Sep. 22, 2023, doi: 10.1109/TIT.2023.3315459.

[34] Z. Wang, M.-A. Badiu, and J. P. Coon, "A framework for characterizing the value of information in hidden Markov models," *IEEE Trans. Inf. Theory*, vol. 68, no. 8, pp. 5203–5216, Aug. 2022.

[35] M. K. Chowdhury Shisher, Y. Sun, and I.-H. Hou, "Timely communications for remote inference," unpublished.

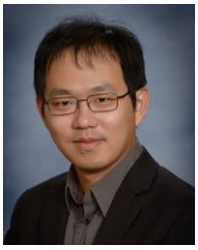
[36] G. Xiong, X. Qin, B. Li, R. Singh, and J. Li, "Index-aware reinforcement learning for adaptive video streaming at the wireless edge," in *Proc. ACM MobiHoc*, 2022, pp. 81–90.
[37] Y. Zou, K. T. Kim, X. Lin, and M. Chiang, "Minimizing age-ofinformation in heterogeneous multi-channel systems: A new partialindex approach," in *Proc. ACM MobiHoc*, 2021, pp. 11–20.
[38] Y. Chen and A. Ephremides, "Scheduling to minimize age of incorrect information with imperfect channel state information," *Entropy*, vol. 23, no. 12, p. 1572, 2021.
[39] D. J. Hodge and K. D. Glazebrook, "On the asymptotic optimality of greedy index heuristics for multi-action restless bandits," *Adv. Appl. Probabil.*, vol. 47, no. 3, pp. 652–667, 2015.
[40] G. Brockman et al., "OpenAI gym," 2016, *arXiv:1606.01540*.
[41] P. D. Grünwald and A. P. Dawid, "Game theory, maximum entropy, minimum discrepancy and robust Bayesian decision theory," *Ann. Statist.*, vol. 32, no. 4, pp. 1367–1433, 2004.
[42] F. Farnia and D. Tse, "A minimax approach to supervised learning," in *Proc. NIPS*, vol. 29, 2016, pp. 4240–4248.
[43] A. P. Dawid, "Coherent measures of discrepancy, uncertainty and dependence, with applications to Bayesian predictive experimental design," Dept. Statist. Sci., Univ. College London., London, U.K.: Rep. 139, 1998.
[44] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, no. 3, pp. 379–423, Jul. 1948.
[45] I. Kadota, A. Sinha, and E. Modiano, "Optimizing age of information in wireless networks with throughput constraints," in *Proc. IEEE INFOCOM*, 2018, pp. 1844–1852.
[46] J. Gittins, K. Glazebrook, and R. Weber, *Multi-Armed Bandit Allocation Indices*. New York, NY, USA: Wiley, 2011.
[47] D. Bertsekas, *Dynamic Programming and Optimal Control: Volume I*, vol. 1. Nashua, NH, USA: Athena Sci., 2017.
[48] R. R. Weber and G. Weiss, "On an index policy for restless bandits," *J. Appl. Probabil.*, vol. 27, no. 3, pp. 637–648, 1990.
[49] D. P. Palomar and M. Chiang, "A tutorial on decomposition methods for network utility maximization," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 8, pp. 1439–1451, Aug. 2006.
[50] R. Andonov, V. Poirriez, and S. Rajopadhye, "Unbounded knapsack problem: Dynamic programming revisited," *Eur. J. Oper. Res.*, vol. 123, no. 2, pp. 394–407, 2000.

Md Kamran Chowdhury Shisher (Member, IEEE) received the B.Sc. degree in electrical and electronic engineering from the Bangladesh University of Engineering and Technology in 2017, and the M.Sc. degree in electrical engineering from Auburn University, Auburn, AL, USA, in 2022, where he is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering. His research interests include information freshness, wireless networks, semantic communications, and machine learning.

Bo Ji (Senior Member, IEEE) received the B.E. and M.E. degrees in information science and electronic engineering from Zhejiang University, Hangzhou, China, in 2004 and 2006, respectively, and the Ph.D. degree in electrical and computer engineering from The Ohio State University, Columbus, OH, USA, in 2012. He is an Associate Professor of Computer Science and the College of Engineering Faculty Fellow with Virginia Tech, Blacksburg, VA,

USA. Prior to joining Virginia Tech, he was an Associate/Assistant Professor with the Department of Computer and Information Sciences, Temple University from July 2014 to July 2020. He was also a Senior Member of the Technical Staff with AT&T Labs, San Ramon, CA, USA, from January 2013 to June 2014. His research interests are in the modeling, analysis, control, and optimization of computer and network systems, such as wired and wireless networks, largescale IoT systems, high-performance computing systems and data centers, and cyber–physical systems. He was a recipient of the National Science Foundation (NSF) CAREER Award in 2017, the NSF CISE Research Initiation Initiative Award in 2017, the IEEE INFOCOM 2019 Best Paper Award, the IEEE/IFIP WiOpt 2022 Best Student Paper Award, and the IEEE TNSE Excellent Editor Award in 2021 and 2022. He has been the General CoChair of IEEE/IFIP WiOpt 2021 and the Technical Program Co-Chair of ACM MobiHoc 2023 and ITC 2021, and he has also served on the editorial boards for the IEEE/ACM TRANSACTIONS ON NETWORKING, *ACM SIGMETRICS Performance Evaluation Review*, IEEE TRANSACTIONS ON

NETWORK SCIENCE AND ENGINEERING, IEEE OPEN JOURNAL OF THE COMMUNICATIONS SOCIETY, and IEEE INTERNET OF THINGS JOURNAL from 2020 to 2022. He is a Senior Member of the ACM.



I-Hong Hou (Senior Member, IEEE) received the Ph.D. degree from the Computer Science Department, University of Illinois at Urbana–Champaign. He is an Associate Professor with the ECE Department, Texas A&M University. His research interests include wireless networks, edge/cloud computing, and reinforcement learning. His work has received the Best Paper Award from ACM MobiHoc 2017 and ACM MobiHoc 2020, and the Best Student Paper Award from WiOpt 2017.

He has also received the C.W. Gear Outstanding Graduate Student Award from the University of Illinois at Urbana–Champaign and the Silver Prize in the Asian Pacific Mathematics Olympiad.

Yin Sun (Senior Member, IEEE) received the B.Eng. and Ph.D. degrees in electronic engineering from Tsinghua University in 2006 and 2011, respectively. He was a Postdoctoral Scholar and a Research Associate with The Ohio State University from 2011 to 2017, and an Assistant Professor with the Department of Electrical and Computer Engineering, Auburn University, Auburn, AL, USA, from 2017 to 2023, where he is currently the Godbold Associate Professor with the Department of Electrical and

Computer Engineering. He coauthored a monograph *Age of Information: A New Metric for Information Freshness*. His research interests include wireless networks, machine learning, semantic communications, age of information, information theory, and robotic control. He is also interested in applying AI and machine learning techniques in agricultural, food, and nutrition sciences. His articles received the Best Student Paper Award of the IEEE/IFIP WiOpt in 2013, the Best Paper Award of the IEEE/IFIP WiOpt in 2019, the runner-up for the Best Paper Award of ACM MobiHoc in 2020, and *Journal of Communications and Networks* Best Paper Award in 2021. He received the Auburn Author Award of 2020, the National Science Foundation CAREER Award in 2023, and was named a Ginn Faculty

Achievement Fellow in 2023. He has been an Associate Editor of the IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING, an Editor of the *Journal of Communications and Networks* and the IEEE TRANSACTIONS ON GREEN COMMUNICATIONS AND NETWORKING, the Guest Editor of five special journal issues, and an organizing committee member of several conferences. He founded the Age of Information Workshop in 2018 and the Modeling and Optimization in Semantic Communications Workshop in 2023.

He is a member of the ACM.