Latency-Distortion Tradeoffs in Communicating Classification Results over Noisy Channels

Noel Teku Sudarshan Adiga Ravi Tandon
Department of Electrical and Computer Engineering
University of Arizona, Tucson, AZ, USA
Email: {*nteku1*, *adiga*, *tandonr*}@arizona.edu

***Abstract*—In this work, the problem of communicating decisions of a classifier over a noisy channel is considered. With machine learning based models being used in variety of time-sensitive applications, transmission of these decisions in a reliable and timely manner is of significant importance. To this end, we study the scenario where a probability vector (representing the decisions of a classifier) at the transmitter, needs to be transmitted over a noisy channel. Assuming that the distortion between the original probability vector and the reconstructed one at the receiver is measured via f-divergence, we study the trade-off between transmission latency and the distortion. We completely analyze this trade-off using uniform, lattice, and sparse lattice-based quantization techniques to encode the probability vector by first characterizing bit budgets for each technique given a requirement on the allowed source distortion. These bounds are then combined with results from finite-blocklength literature to provide a framework for analyzing the effects of both quantization distortion and distortion due to decoding error probability (i.e., channel effects) on the incurred transmission latency.**

Our results show that there is an interesting interplay between source distortion (i.e., distortion for the probability vector measured via f-divergence) and the subsequent channel encoding/decoding parameters. We observe that the source distortion can be optimized for each quantization technique to attain a minimum latency. Our results also indicate that sparse lattice-based quantization is the most effective at minimizing latency for low end-to-end distortion requirements across different parameters and works best for sparse, high-dimensional probability vectors (i.e., high number of classes). To corroborate our framework, we use the quantization techniques on predictions made on real datasets and send them through a simulated channel. We use the metric of 'relative accuracy' to measure how often the class assigned with the highest probability by the classifier at the transmitter is correctly identified after transmission. Our results indicate that the lattice-based techniques require significantly smaller blocklengths than uniform quantization (subsequently incurring smaller latencies) but can still provide a comparable performance to uniform quantization.

***Index Terms*—Low-Latency, Quantization, Finite blocklength**

I. Introduction

This work was supported by NSF grants CAREER 1651492, CCF-2100013, CNS-2209951, CNS-1822071, CNS-2317192, and by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing under Award Number DE-SC-ERKJ422, and NIH Award R01-CA261457-01A1. Parts of this paper were presented at the 2024 IEEE International Conference on Communications [1]. The authors are with the Department of Electrical and Computer Engineering, The University of Arizona, Tucson, AZ 85721 USA (E-mail: nteku1@arizona.edu; adiga@arizona.edu; tandonr@arizona.edu).

IN recent years, machine learning (ML) has been increasingly applied to time-sensitive applications, including Vehicle to Vehicle (V2V) and Vehicle to Infrastructure (V2I) communications. These applications require reliable and rapid data transmission for tasks such as trajectory prediction [2] and lane change detection [3]. Similarly, this need for reliable, fast communication extends to other domains like internet of things (IoT) and edge computing. Coinciding with the increasing use of ML in low-latency applications, there has also been a growing body of work on context-dependent low-latency communications; which includes semantic communications [4]–[7], ultra-reliable low latency communications (URLLC) [8], [9], and joint source channel coding [10]–[13].

Semantic communication generally focuses on sending *context dependent* features/decisions dependent on the data to the receiver (rather than the entire raw message) [7]. In doing so, the amount of bits required for transmission is often reduced [6]. For example, in [14], a transformer-based network was used to learn/transmit semantic features of sentences and decode the received features to ensure that the original meaning of the sentences were preserved. In [15], an approach to modeling the length of a semantic message and its distortion based on noise due to the model and the channel is presented along with masking strategies that can be applied before transmission. [16], [17] present rate-distortion approaches for semantic communications for general block-wise distortion functions. The focus of URLLC is to design protocols in order to transmit low-data rate (short packets) with high reliability (low probability of error) within a small latency [9]. A rate-distortion analysis is also performed in [18] for short control packets, assuming transmissions are being made to a remote agent, where the distortion measures considered are quantization error and the freshness of the data (age of information); however, this analysis is done under the assumption of noiseless channels.

Related Works and Main Contributions: In this paper, we focus on the following problem: a transmitter wishes to send a probability vector (e.g., representing the decisions of a ML based classifier) to a receiver over a noisy channel. Our objective in focusing on transmitting probability vectors is to observe if there are any properties of such a vector that can be exploited for obtaining reductions in latency when transmitting while still preserving the decision of the classifier. When

This article has been accepted for publication in IEEE Transactions on Communications. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2024.3484939

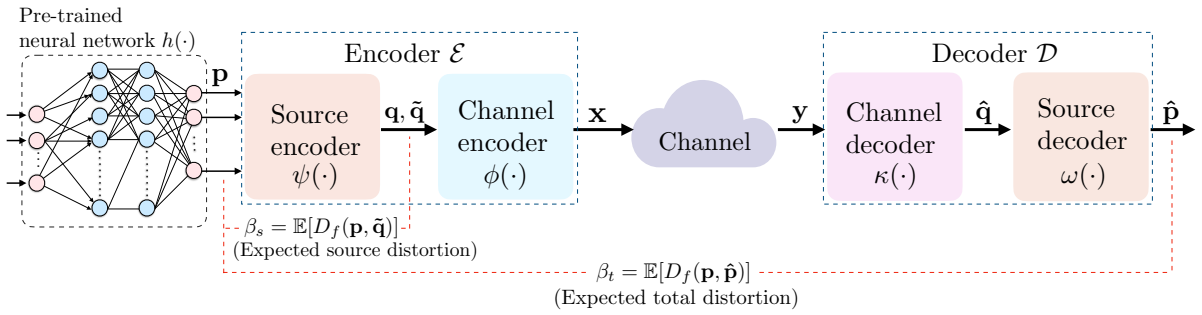


Fig. 1. End-to-End block diagram for communicating classifier decisions (probability vector) over a noisy channel.

using datasets with a high number of classes (e.g. CIFAR-100, Imagenet-1k), for example, there will be many entries in the vector that do not add any substantial information regarding the classifier's decision. In such a scenario, a vector containing a small number of a classifier's highest predictions may be sufficient for identifying a classifier's decision compared to sending the entire vector. Results obtained from classifiers operating on real datasets are provided later in the paper that further support this idea. Additionally, transmitting probability vectors enables the use of specific quantization techniques that can be efficient. One such technique, which is considered in this work, is a lattice based quantization proposed in [19], where a given probability distribution is fitted to its nearest match on a certain finite-dimensional lattice. This algorithm was also applied in [20] for the application of image retrieval. Descriptors representing the histogram of gradients of an image were quantized using the technique in [19] and sent to a server hosting a database of images; the gradient descriptors of the desired images from the server then underwent the same quantization and were sent back to the transmitter. The algorithm itself, as noted in [19] is similar (but more specialized) to the algorithm presented by Conway and Sloane in [21], which showed how to find the nearest representative point on various lattices for a given input vector. This problem can also be viewed within the umbrella of semantic communication and joint source channel coding. Transmitting the results of a classification task incurs lower latency/overhead compared to sending a compressed form of the data required for classification at the receiver. It also enables the receiver to quickly execute tasks that depend on knowing the classification results, which is essential to conducting goal-oriented communications [5]. Additionally, this problem falls under the umbrella of JSCC as its objective is to attain low-latency transmissions by operating in the finite blocklength regime [11].

The main new elements herein are two fold: we measure utility of the reconstruction of the probability vector in terms of statistical divergence measures; and secondly, we simultaneously want to minimize the transmission latency over the noisy channel. We note that there has been prior work on quantizing probability distributions, including [19], [22]–[25]. In particular, [24], [25] investigated quantizing probability distributions in order to minimize Kullback-Leibler (KL)-divergence by performing a non-linear operation and then using uniform quantization. However, the existing works did not study the scenario when a probability vector has to be transmitted through a noisy channel, and what would be the right quantization strategy/parameters if the goal is to minimize latency. By considering the distortion introduced by the channel and quantization, we aim to analyze the trade off between the end-to-end distortion of the system and the incurred transmission latency. Additionally, a similar framework considering quantization noise was introduced in [26], but focused on transmitting control signals and ensuring the stability of the assumed control system rather than quantizing/transmitting probability vectors. There have also been works such as [27]–[29] that look at relating finite blocklength analysis with latencies but do not consider quantization noise.

Our main contributions are as follows:

- *In-depth investigation of quantization techniques for classification results:* The performance of uniform, lattice, and sparse lattice-based quantization techniques are investigated with respect to balancing the trade-off between latency and end-to-end distortion. We show that the lattice-based methods are more efficient than the baseline uniform quantization as they require less complexity and make use of the properties of the probability vector to require fewer bits. The sparse-lattice based technique is proposed to employ the assumed lattice-based quantization technique on only a few of the highest probabilities of the vector. This amount should be determined such that a large portion of the mass of the vector is represented, which is investigated on predictions from classifiers on real datasets; specifically, CIFAR-100 & Imagenet-1K. We provide results bounding the necessary bit budgets under each technique to satisfy a requirement on the allowable source distortion (Lemmas 2-4). Our results show the expected trend that to ensure a lower source distortion when quantizing the probability vector, a higher bit budget is needed for each of the assumed techniques. For a probability vector of length 50 classes and the same source distortion, for example, our results show that sparse-lattice based quantization incurs a reduction in bit requirement of approximately 96% and 80% with respect to uniform and lattice-based quantization.
- *Latency-Distortion trade-off analysis:* We derive a relationship (Lemma 5) between the source distortion incurred for each of the afroementioned quantization techniques and the decoding error probability (i.e. accounting

for distortion caused by noisy channel effects) to obtain a bound on the end-to-end distortion between the received and transmitted vectors. By incorporating these two sources of distortion, the subsequent blocklength under these parameters can be obtained and used to calculate the transmission latency (Theorem 1). Our results show that by using the proposed framework, an optimized source distortion can be found that achieves a minimal latency for different levels of end-to-end distortion. In doing so, this also enables us to extend our framework to fading channels (Theorems 2 & 3).

- *Application to noisy channels:* We provide a comprehensive set of simulation results to validate the proposed framework. Specifically, we study the trade-off between accuracy, latency and distortion while varying parameters of the framework; such as, channel conditions (i.e. SNR), source distortion, and the length of the probability vector (i.e. number of classes). We first report results assuming additive white gaussian noise (AWGN) and fading channels using results from the literature on finite blocklength, focusing on the latency-distortion tradeoff. For a probability vector of length 100 classes and the same end-to-end distortion, for example, our results show that the sparse-lattice based quantization can incur a latency reduction of approximately 97% and 85% with respect to uniform and standard lattice-based quantization for the AWGN channel. Our results indicate that sparse lattice-based quantization is the most effective at minimizing latency for low end-to-end distortion requirements across different parameters. Specifically, the results indicate that sparse lattice-based quantization works best for sparse, high-dimensional probability vectors (i.e. high number of classes). We then present results showing the tradeoff between accuracy, latency, and distortion through simulated AWGN channels by quantizing predictions made on real datasets. The metric of 'relative accuracy' is used to measure how often the receiver can determine which class was originally given the highest probability by the classifier at the transmitter. Finally, as an application, results are presented on a collaborative scenario where multiple transmitters pass their noisy observations of an input to a classifier and the subsequent classification results (probability vectors) are transmitted through multiple channels of different quality. The receiver must employ a fusion strategy on the received vectors to decide which class the classifier would have assigned the highest probability at the transmitter if given a noiseless version of the input.

The paper is structured as follows: Section II presents the system model studied in this work; Section III presents results analyzing the distortion incurred with each of the quantization techniques and details our framework for analyzing the latency-distortion tradeoff using these techniques for AWGN and fading channels; Section IV presents results from simulations and experiments; Section V concludes the paper and proposes future work. The proofs for the technical results are presented in the Appendix.

II. System Model

We consider the scenario illustrated in Fig. 1: a pre-trained classifier (e.g., a neural network), denoted as $h(\cdot)$, is used for a k-class classification problem and is situated at a transmitter. The output classification probabilities are represented as $\mathbf{p} = [\mathbf{p}[1], \mathbf{p}[2], \cdots, \mathbf{p}[k]]^\top$, where $\mathbf{p} \in \mathbb{R}^{k\times 1}$. Let $\hat{\mathbf{p}} = [\hat{\mathbf{p}}[1], \hat{\mathbf{p}}[2], \cdots, \hat{\mathbf{p}}[k]]^\top$ denote the estimated classifier output at the receiver. In this paper, we measure the distortion between $\mathbf{p}$ and $\hat{\mathbf{p}}$ via f-divergence, defined as

$$D_{\mathrm{f}}(\mathbf{p}, \hat{\mathbf{p}}) = \sum_{i=1}^{k} f\left(\frac{\mathbf{p}[i]}{\hat{\mathbf{p}}[i]}\right) \hat{\mathbf{p}}[i]. \tag{1}$$

The transmitter's goal is to communicate the probability vector $\mathbf{p}$ within a latency budget of $T_{\max}$ with maximum total expected distortion β_t, i.e., $\mathbb{E}(D_{\mathrm{f}}(\mathbf{p}, \hat{\mathbf{p}})) \leq \beta_t$, where the expectation is over the noisy channel realizations. We next describe the main components (source/channel encoder/decoder(s)): a source encoder $\psi(\cdot)$ quantizes the probability vector $\mathbf{p}$, such that $\mathbf{q} = \psi(\mathbf{p})$. The lossy compression caused by quantization results in source distortion, denoted by β_s. The total number of bits required by $\mathbf{q}$, given the source distortion, is represented as $J(\beta_s)$, where $J(\cdot)$ is a function of β_s. We note that based on the quantization technique, $\mathbf{q}$ may not necessarily be a probability vector. In this scenario, we normalize the values in $\mathbf{q}$ to obtain the corresponding probability vector $\tilde{\mathbf{q}}$ after source encoding, where $\tilde{\mathbf{q}}[i] = \mathbf{q}[i] / \sum_{i=1}^{k} \mathbf{q}[i]$ and $\tilde{\mathbf{q}} \in \mathbb{R}^{k\times 1}$. The source distortion β_s is quantified as $\beta_s = D_{\mathrm{f}}(\mathbf{p}, \tilde{\mathbf{q}})$. We use the channel encoder $\phi(\cdot)$ to generate the n-length channel input $\mathbf{x} = \phi(\mathbf{q})$, where $\mathbf{x} = [\mathbf{x}[1], \mathbf{x}[2], \cdots, \mathbf{x}[n]]^\top$ and $\mathbf{x} \in \mathcal{X}^n$. Let $\mathcal{E}$ denote the source and channel encoder pair. We consider a fading channel, where the channel output is given by $\mathbf{y}[i] = \mathbf{h}[i]\mathbf{x}[i] + \mathbf{z}[i]$, for all $i \in [n]$; where $\mathbf{y} \in \mathbb{R}^{n\times 1}$, the channel fading gains are given by $\mathbf{h} = [\mathbf{h}[1], \mathbf{h}[2], \cdots, \mathbf{h}[n]]^\top$ with $\mathbf{h} \in \mathbb{R}^{n\times 1}$, and the AWGN noise vector is given by $\mathbf{z} = [\mathbf{z}[1], \mathbf{z}[2], \cdots, \mathbf{z}[n]]^\top$ with $\mathbf{z} \in \mathbb{R}^{n\times 1}$. We also consider an AWGN channel which can be obtained from the above model by setting $\mathbf{h}[i] = 1 \ \forall \ i \in n$. The signal-to-noise ratio (SNR) of the channel for a bandwidth B_0 Hz, is defined as $\gamma_0 = \frac{P}{N_0}$, where P denotes the signal power and N_0 denotes the noise power. To simulate using the same transmit powers at different bandwidths, as done in [8], we define the operational SNR for a channel of bandwidth B Hz as $\gamma = \frac{\gamma_0 B_0}{B}$ where $\frac{B_0}{B}$ acts as a scaling factor for relating different channel conditions.

We denote the decoding error probability by $\epsilon^*(n)$, where $\epsilon^*(n) \in [0, 1]$. At the receiver, we consider a channel decoder, denoted by $\kappa(\cdot)$, such that $\hat{\mathbf{q}} = \kappa(\mathbf{y})$. Subsequently, we consider the source decoder $\omega(\cdot)$ and a normalization operation to obtain an estimate of the classifier probabilities, given by $\hat{\mathbf{p}} = \omega(\hat{\mathbf{q}})$. Let $\mathcal{D}$ denote the source and channel decoder pair.

The channel noise, in addition to the source distortion, contributes to the total end-to-end distortion. Given a specific SNR, it is possible to vary the source distortion β_{s} to achieve a maximum total expected distortion of β_{t}. In other words, we have $\beta_s \in [0, \beta_{\mathrm{t}}]$. This choice will also affect the incurred

transmission latency; given a bandwidth of B Hz, the time required to transmit an n-length vector $\mathbf{x}$ is calculated as:

$$T(\mathcal{E}, \mathcal{D}) = \frac{n}{2B}. \tag{2}$$

In this paper, we focus on understanding the tradeoff between latency and distortion for the task of communicating probability distributions. Specifically, given the channel statistics (e.g., bandwidth, SNR) and desired maximum latency $T_{\max}$, the optimal distortion can be defined as follows:

$$D^*(T_{\max}) \triangleq \min_{(\mathcal{E},\mathcal{D})} \quad \beta_t(\mathcal{E}, \mathcal{D}), \quad \text{s.t.} \;\; T(\mathcal{E}, \mathcal{D}) \leq T_{\max}. \tag{3}$$

Alternatively, we can fix the maximum permissible distortion $\beta_{\max}$, and minimize the total latency T over encoder-decoder pairs as

$$T^*(\beta_{\max}) \triangleq \min_{(\mathcal{E},\mathcal{D})} \quad T(\mathcal{E}, \mathcal{D}), \quad \text{s.t.} \;\; \beta_t(\mathcal{E}, \mathcal{D}) \leq \beta_{\max}. \tag{4}$$

In the lemma stated next (proof is presented in the Appendix), we show that the optimal latency $T^*(\beta_{\max})$ is a convex non-increasing function of the total distortion $\beta_{\max}$; and likewise, we show that the minimal distortion $D^*(T_{\max})$ is a convex non-increasing function of $T_{\max}$.

Lemma 1. *$T^*(\beta_{max})$ is convex non-increasing function of β_{max}. $D^*(T_{max})$ is convex non-increasing function of T_{max}.*

III. Main Results & Discussion

In this section, we present the framework for analyzing the latency-distortion tradeoff. We begin by assuming a noiseless channel and uniform quantization as the source encoder (i.e., transforming $\mathbf{p}$ to $\mathbf{q}$) and analyze the corresponding source distortion (Lemma 2). We perform a similar analysis for lattice and sparse lattice-based quantization techniques to analyze the source distortion for a noiseless channel (Lemma 3 & Lemma 4). We then incorporate and analyze the impact of channel noise on the end-to-end distortion (Lemma 5). Subsequently, we use results on finite-blocklength capacity, which allow us to connect latency with the overall distortion. This, in turn, also leads to an explicit optimization (Theorem 1), which can be solved to trade latency with distortion. We then extend this result to account for fading channels with and without CSI (Theorems 2 & 3).

A. Quantizing Classifier Probabilities

1) Uniform Quantization (UQ): Suppose we have a total budget of J bits to quantize the k-dimensional probability vector $\mathbf{p}$. Under uniform quantization (UQ), we use $j = \lfloor J/k \rfloor$ bits to quantize each element $\mathbf{p}[i], i = 1, 2, \ldots, k$. We denote $\mathbf{q}[i]$ as the resulting quantized output. Note that $\mathbf{q}$ may not necessarily be a probability vector. We can however, normalize it as $\tilde{\mathbf{q}}[i] = \frac{\mathbf{q}[i]}{\sum_{\ell=1}^{k} \mathbf{q}[\ell]}$, for $i = 1, \ldots, k$. Our objective is to minimize the f-divergence between $\mathbf{p}$ & $\hat{\mathbf{p}}$; in the noiseless scenario, which would be equivalent to minimizing $D_{\mathrm{f}}(\mathbf{p}, \tilde{\mathbf{q}})$, as β_s would be the only distortion present. When $f(x) = \frac{1}{2}|x-1|$, f-divergence results in total variation (TV): $D_{\mathrm{f}}(\mathbf{p}, \mathbf{q}) = D_{\mathrm{TV}}(\mathbf{p}, \mathbf{q}) = \frac{1}{2}\sum_{i=1}^{K} |\mathbf{p}[i] - \mathbf{q}[i]|$ [30]. The next lemma shows a sufficient condition on the quantization budget to achieve a source distortion of β_s.

Lemma 2. *For a k-class classification problem, if the total uniform quantization (UQ) budget satisfies*

$$J_{UQ} \geq 2k \cdot \log_2\left(\frac{k}{\beta_s}\right), \tag{5}$$

then $D_{TV}(\mathbf{p}, \tilde{\mathbf{q}}) \leq \beta_s$.

Remark 1. *(Impact of normalization) The proof of Lemma 2 is non-trivial due to the nature of the vectors involved. While $\mathbf{p}$ is a probability vector, the corresponding quantized $\mathbf{q}$ may not be a probability vector. To apply the statistical f-divergence measure, we normalize the entries of $\mathbf{q}$ by their sum, i.e., $S = \sum_i \mathbf{q}[i]$. However, this normalization operation makes the analysis of bounding the f-divergence challenging. The proof above overcomes this issue, by first assuming that the number of bits j is of the form $j = \log(k/2\alpha)$, and then we are able to bound the sum S as $S \in [1-\alpha, 1+\alpha]$. This allows us to determine the number of bits required to achieve a desired source distortion β_s.*

2) Lattice-based Quantization (LQ): There are a few disadvantages when using UQ. First, UQ does not exploit the fact that the vector being compressed is a probability distribution. There are more efficient methods that can further reduce the number of bits required for quantization by exploiting this property. Additionally, as the number of classes k increases, the length of $\mathbf{p}$ will increase, leading to a significant increase in the number of bits required to satisfy the source distortion requirement as shown in Lemma 2. Also, as noted in Remark 1, UQ requires an additional normalization step which complicates deriving a bound on the source distortion between $\mathbf{p}$ and $\tilde{\mathbf{q}}$. Subsequently, we now consider the algorithm presented in [19], which presents a lattice-based approach for quantizing probability distributions. The algorithm uses a lattice to represent a set of k-length probability distributions that is a subset of the k-dimensional probability simplex (i.e. the set of all possible k-length probability vectors $\mathcal{A}_k = \{[\mathbf{q}[1], ..., \mathbf{q}[k]] \in \mathbb{Q}^k \mid \sum_i \mathbf{q}[i] = 1\}$). The probability vectors in the lattice are defined to have the property that each of their elements must have the same denominator ℓ, which is a positive integer set by the user. Subsequently, each element in the probability vector must be of the form $\mathbf{q}[i] = \frac{\mathbf{b}[i]}{\ell}$, where $\mathbf{b}[i]$ are also positive integers. Because $\mathbf{q}$ must sum to 1, this implies that $\sum_i \mathbf{b}[i] = \ell$. Denoting the lattice as Q_ℓ, the formal structure for the lattice is given as follows [19]:

$$Q_\ell = \{[\mathbf{q}[1], ..., \mathbf{q}[k]] \in \mathbb{Q}^k \mid \mathbf{q}[i] = \frac{\mathbf{b}[i]}{\ell}, \sum_i \mathbf{b}[i] = \ell\}. \tag{6}$$

From (6), we can see that $Q_\ell \subseteq \mathcal{A}_k$ and if a probability distribution satisfies (6), it is a point on Q_ℓ. The algorithm's objective is to find the point (i.e. probability distribution) on Q_ℓ closest, under an assumed distance metric, to a given probability distribution $\mathbf{p}$. We denote the resulting probability distribution chosen from Q_ℓ as $\mathbf{q}_{\mathrm{LQ}}(\mathbf{p})$. The procedure for this method is summarized in Algorithm 1. First, an initial guess of the nearest distribution based on $\mathbf{p}$ and ℓ is made using a simple mapping. If the mapping immediately results

Algorithm 1 Lattice-based Quantization (LQ) [19]

Inputs: $\mathbf{p}$, Q_ℓ
Compute $\mathbf{b}^{'}[i] = \lfloor \ell \mathbf{p}[i] + \frac{1}{2} \rfloor, \ell' = \sum_i \mathbf{b}^{'}[i]$
if $\ell^{'} = \ell$ **then**
  Done
else
  Calculate $\zeta[i] = \mathbf{b}^{'}[i] - \ell \mathbf{p}[i]$ and sort in increasing order.
  if $\ell^{'} - \ell > 0$ **then**
    Decrease $|\ell' - \ell|$ values with largest $\zeta[i]$ in $\mathbf{b}^{'}[i]$ by 1
  else if $\ell' - \ell < 0$ **then**
    Increase $|\ell' - \ell|$ values with smallest $\zeta[i]$ in $\mathbf{b}^{'}[i]$ by 1
  end if
end if
Compute lexicographic index to represent $\mathbf{b}[1], ..., \mathbf{b}[k]$ as follows:

$$\xi(\mathbf{b}[1], ..., \mathbf{b}[k]) = \sum_{j=1}^{\ell-2} \sum_{i=0}^{\mathbf{b}[j]-1} \binom{u}{k-j-1} + \mathbf{b}[k-1] \quad (7)$$

where $u = \ell - i - \sum_{a=1}^{j-1} \mathbf{b}[a] + k - j - 1$.

in a probability distribution on Q_ℓ the algorithm is complete; otherwise, updates are made to the guess based on the observed error to push it to the nearest point on Q_ℓ. Thus, by mapping $\mathbf{p}$ to one of the available distributions on Q_ℓ, lattice-based quantization (LQ) requires significantly less complexity compared to UQ. As an example, assume that we are given a probability vector $\mathbf{p} = [0.18, 0.52, 0.3]$ (meaning that $k = 3$) and $\ell = 5$. This means that Q_5 consists of all probability vectors of length 3 whose entries have 5 as a denominator. Examples of candidate probability vectors in Q_5 include $[\frac{1}{5}, \frac{2}{5}, \frac{2}{5}]$, $[\frac{1}{5}, \frac{1}{5}, \frac{2}{5}]$, and $[\frac{3}{5}, \frac{2}{5}, 0]$. Applying the initial mapping of $\mathbf{p}$ as shown in Algorithm 1, results in an initial guess of $\mathbf{b}' = [1, 3, 2]$. However, this means that we have $\ell' = \sum_i \mathbf{b}'[i] = 6$. Because $\ell' \neq \ell$, we must perform updates to push this guess closer to Q_ℓ. We first calculate how far away the guess is from an actual point on Q_ℓ by performing $\zeta[i] = \mathbf{b}[i] - \ell \mathbf{p}[i]$, which results in $\zeta = [0.1, 0.4, 0.5]$. Because $\ell' - \ell = 1$, we must decrement the element in $\mathbf{b}'$ with the largest $\zeta[i]$ by 1. From this example, we can see that the third element in $\mathbf{b}'$ has the largest error; decrementing it gives us $\mathbf{b}' = [1, 3, 1]$, which results in $\mathbf{q}_{\text{LQ}}(\mathbf{p}) = [\frac{1}{5}, \frac{3}{5}, \frac{1}{5}]$.

We note that [19] uses the L_1, L_2 and L_∞ norm to report worst-case distance metrics between $\mathbf{p}$ and $\mathbf{q}_{LQ}(\mathbf{p})$. By noting that the L_1 norm is equivalent to $2D_{TV}(\mathbf{p}, \mathbf{q}_{LQ}(\mathbf{p}))$, based on [19] the maximum source distortion between $\mathbf{p}$ and $\mathbf{q}_{LQ}(\mathbf{p})$ is as follows:[1]:

$$D_{TV}(\mathbf{p}, \mathbf{q}_{\text{LQ}}(\mathbf{p})) = \frac{k}{4\ell}. \quad (8)$$

Two observations can be made from (8). First, for high dimensional lattices, the resulting source distortion decreases, meaning that the distributions on the lattice are closer representations of $\mathbf{p}$. Second, the guarantee on the source distortion becomes looser as the length of $\mathbf{p}$ increases. This intuitively makes sense because each distribution on the lattice will have a longer length but still need to satisfy the summation constraint in (6), leading to distributions that are more distinct from $\mathbf{p}$. Once $\mathbf{q}_{LQ}(\mathbf{p})$ is determined, its corresponding index is calculated and transmitted. The number of bits required to send this index under this method is as follows [19]:

$$J_{\text{LQ}} = \left\lceil \log_2 \binom{\ell + k - 1}{k - 1} \right\rceil. \quad (9)$$

The next lemma shows the number of bits required under this quantization technique to attain a source distortion β_s

Lemma 3. *For a k-class classification problem, if the total lattice-based quantization (LQ) budget under Algorithm 1 satisfies*

$$J_{LQ} \geq \left\lceil \log_2 \binom{\ell + k - 1}{k - 1} \right\rceil, \quad (10)$$

where $\ell = \left\lceil \frac{k}{4\beta_s} \right\rceil$, then $D_{TV}(\mathbf{p}, \mathbf{q}) \leq \beta_s$.

Complexity Comparison of Various Schemes: Under LQ, any updates that may be needed after the projection of $\mathbf{p}$ onto Q_ℓ are only performed once on the values that contribute to the largest error (see Algorithm 1); thus, these updates require at most $\mathcal{O}(k)$ complexity. Once the vector from Q_ℓ is determined, its lexicographic index is calculated as summarized in Algorithm 1, which can require significant complexity. However, this can be reduced by pre-saving the binomical coefficients which would only require storing $\mathcal{O}(k\ell)$ terms instead of $\mathcal{O}(k^\ell)$ and would only need $\mathcal{O}(\ell)$ operations to determine the index [20]. As stated earlier, unlike LQ, UQ does not exploit any property of the probability vector to reduce the complexity of quantization. As k increases UQ will need significantly more bits to represent the probability vector. An increase in k will also cause LQ to incur high computational complexity, which motivates us to propose a sparse version of the algorithm later in this work. Finally, we pose the last disadvantage UQ has in the following remark.

Remark 2. *It can be observed from Lemma 3 that the required bits for LQ has an expression similar to the required bits for UQ as shown in Lemma 2. However, unlike UQ, LQ does not require an additional normalization operation, which reduces the complexity of the derivation.*

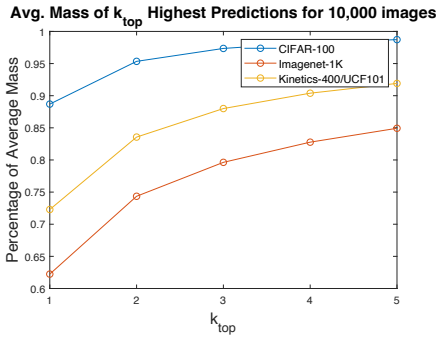
3) Sparse Lattice-based Quantization (SLQ): For large values of k, it may be desired to only send a certain number of the top highest predictions in the k-dimensional probability vector due to many elements of the vector being very close to 0. To accommodate this, we now propose a sparse version of the algorithm presented in [19]. The motivation for this method comes from the notion that for high dimensional datasets (i.e. high k), a large portion of the mass of a classifier's decisions is concentrated in the k_{top} highest predictions (i.e. $\sum_{i \in k_{\text{top}}} \mathbf{p}[i] \geq 1 - \delta$, where $0 < \delta < 1$ represents the mass of the $k - k_{\text{top}}$ lowest probabilities). As a case study, Figures 2b plots the average mass of the k_{top} highest predictions outputted by different neural network architectures on various datasets. The networks and datasets evaluated used to generate the figure are summarized in Figure 2a. Imagenet-1K [40] and

[1] [19] proves the maximum L_1 distance as $\frac{1}{\ell} \frac{2a(k-a)}{k}$, where $a = \left\lfloor \frac{k}{2} \right\rfloor$. By assuming even values of k, the simplified expression in (8) is obtained.

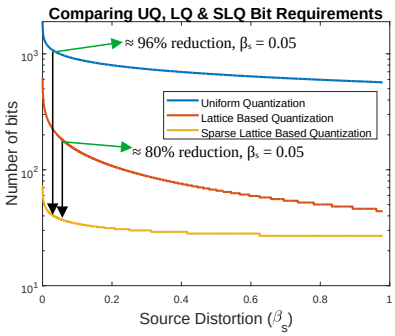
Summary of Network Architectures & Datasets In Figure 2b

Network Architecture	Inferenced Dataset	k (Number of classes)
VGG-16 [32]-[35]	CIFAR-100	100
Resnet-50 [36]-[37]	Imagenet-1k	1000
I3D [38]-[39] (pretrained on Kinetics-400 dataset)	UCF-101	400

(a)



(b)



(c)

Fig. 2. (a) Table summarizing the pre-trained network architectures and datasets used to generate Figure 2b. (b) Percentage of average mass of k_{top} highest predictions from different pre-trained networks and datasets across $10,000$ test images. (c) Bit requirements for UQ, LQ, and SLQ (with $k_{\text{top}} = 5$ & $\delta = 0.00001$) for different class sizes as a function of source distortion.

CIFAR-100 [41] are variations of the Imagenet and CIFAR-10 image datasets that have been frequently used in the machine learning literature, containing 1000 and 100 classes respectively. Kinetics-400 [42] and UCF-101 [43] are datasets that have been generated for classifying human actions present in videos, with each having 400 and 101 classes respectively. The figure indicates that as k_{top} increases gradually, the average mass of the probability vector encapsulated by these k_{top} values also increases. This is beneficial as it further shows that transmitting the whole probability vector may not be required to accurately identify the classifier's decision on an image.

Under sparse-lattice based quantization (SLQ), the k_{top} highest values of the probability vector $\mathbf{p}$ are chosen to constitute the sparse vector $\mathbf{q}$. However, $\mathbf{q}$ does not constitute a probability vector and must be normalized, which is denoted as $\bar{\mathbf{q}}$. Algorithm 1 is then used to perform LQ on the normalized sparse vector; the resulting vector is denoted as $\mathbf{q}_{\text{SLQ}}(\mathbf{p})$. The positions of the k_{top} highest predictions also need to be transmitted for the receiver to know which classes the probabilities correspond to. We quantize the set of positions of the k_{top} highest values by generating a set of k bits where a 1 is used for an index if it is one of the k_{top} highest probabilities and a 0 is used otherwise. Subsequently, the k bits needed to represent the indices can be lower bounded by $\left\lceil \log_2 \binom{k}{k_{\text{top}}} \right\rceil$. The total number of bits needed to send the index for $\mathbf{q}_{\text{SLQ}}(\mathbf{p})$ is given as $\left\lceil \log_2 \binom{\ell+k_{\text{top}}-1}{k_{\text{top}}-1} \right\rceil$. This has a similar form to the number of bits needed for the regular LQ given in (9). It can be observed that using this procedure introduces two sources of distortion: normalization and lattice-based quantization. The next lemma shows a lower bound on the required number of bits for this technique to satisfy this more stringent restriction on the source distortion.

Lemma 4. *For a k-class classification problem, if the total sparse lattice-based quantization (SLQ) budget satisfies*

$$J_{SLQ} \geq \left\lceil \log_2 \binom{k}{k_{top}} \right\rceil + \left\lceil \log \binom{\ell + k_{top} - 1}{k_{top} - 1} \right\rceil, \qquad (11)$$

where $\ell = \left\lceil \frac{k_{top}}{4(\beta_s - \delta)} \right\rceil$ & $\sum_{i \notin k_{top}} \mathbf{p}[i] \leq \delta$, then $D_{TV}(\mathbf{p}, \mathbf{q}_{SLQ}(\mathbf{p})) \leq \beta_s$.

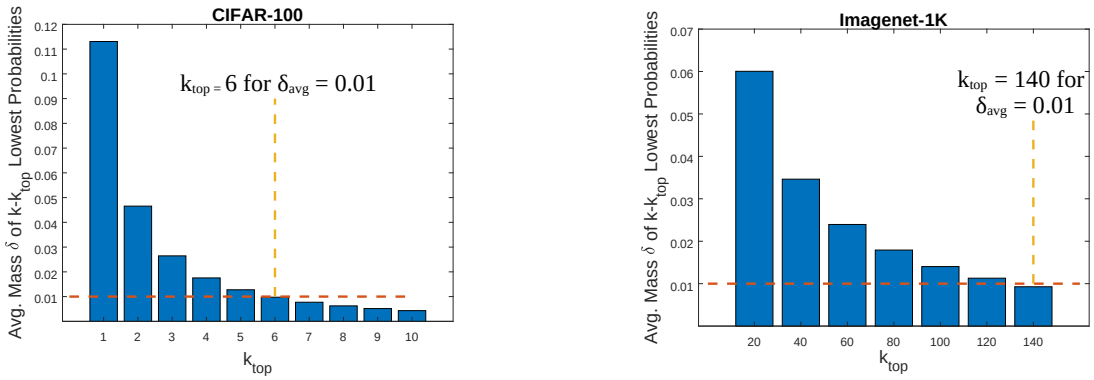
Remark 3. *Despite the additional term in Lemma 4, compared to that of Lemma 3, because fewer bits are sent, the number of bits required under SLQ should be less compared to its standard counterpart for large k-dimensional vectors. Additionally, we also see that compared with the standard LQ, the choice of ℓ is also dependent on the mass encapsulated by the $k - k_{top}$ lowest values in the probability vector.*

Comparison of Bounds: To develop an intuition as to how the bounds in Lemmas 2-4 compare with each other, Figure 2c plots each of them as a function of the source distortion β_s. The figure was generated assuming $k = 50$ classes and for SLQ $k_{\text{top}} = 5$. The figure indicates that as the allowable amount of source distortion increases, the number of bits required for UQ, LQ, and SLQ decreases. The figure also indicates that for a relatively high number of classes, SLQ requires the lowest number of bits. Looking at $\beta_s = 0.05$, for example, SLQ incurs a reduction in bit budget of approximately 96% and 80% with respect to UQ and LQ. However, it can also be observed that LQ requires significantly fewer bits compared to UQ. Observing Lemmas 2-3 for large k, the number of bits for UQ and LQ an be approximated as $\mathcal{O}(k \log(\frac{k}{4\beta_s}))$ & $\mathcal{O}(k \log(\frac{1}{4\beta_s}))$. This implies that LQ requires approximately $\mathcal{O}(\log(k))$ fewer bits compared to UQ for the same β_s.

It is worth noting that for high β_s, the figure indicates that the number of bits required for LQ starts to approach that of SLQ. However, this phenomenon is intuitive, because when a high amount of source distortion is allowed, this implies that fewer bits are needed as the requirement to meet the total distortion constraint is placed more on the channel encoder/decoder rather than the source encoder. Thus, we care more about the performance of the quantization schemes for low β_s, and Figure 2c indicates that SLQ significantly outperforms UQ and LQ in this regime.

Selection of k_{top}: The latency incurred by using SLQ is contingent on the choice of k_{top}. However, k_{top} cannot be chosen arbitrarily because the choice of k_{top} affects the average mass of the probability vector represented in the transmitted quantized vector. Figure 3 shows the average mass of the $k - k_{\text{top}}$ lowest probabilities (denoted as δ_{avg}) for predictions made on CIFAR-100 and Imagenet-1K. As k_{top} increases, and

This article has been accepted for publication in IEEE Transactions on Communications. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2024.3484939



(a) (b)

Fig. 3. Average mass of $k - k_{\text{top}}$ lowest probabilities for different values of k_{top} for classifications made on 10,000 images of (a) CIFAR-100 (b) Imagenet-1K.

more of the mass of the probability vector is subsequently represented by the k_{top} highest values, we see that the average mass of the $k - k_{\text{top}}$ values decreases for both datasets. Because of the higher number of classes in Imagenet-1K, a larger choice of k_{top} is needed to reduce δ_{avg} compared to CIFAR-100. If a requirement was given for $\delta_{\text{avg}} < 0.01$, for example, then for CIFAR-100 and Imagenet-1K this would require approximately a k_{top} of 6 & 140 respectively.

B. Tradeoff Between Latency & End-to-End Distortion

Impact of Channel Noise & Decoding Error: We now incorporate the effect of channel noise and decoding errors in the analysis of the end-to-end distortion. The next lemma shows a bound on the overall expected distortion (expectation is over the channel noise realizations) if the source distortion is bounded by β_s, and the decoding error probability is given by $\epsilon^*(n)$.

Lemma 5. *For a given source distortion β_s and decoding error probability $\epsilon^*(n)$, the overall expected distortion is upper bounded as follows:*

$$\mathbb{E}[D_{TV}(\mathbf{p}, \hat{\mathbf{p}}(\kappa(\mathbf{y})))] \leq (1 - \epsilon^*(n))\beta_s + \epsilon^*(n). \tag{12}$$

Remark 4. *An observation from Lemma 5 is that there is no explicit dependence on the specific quantization technique. The bound on overall distortion is only dependent on the source distortion β_s and decoding error probability introduced via $\epsilon^*(n)$. This indicates that this framework can work generally for different quantization techniques (for instance, one could replace uniform quantization with some other sophisticated non-uniform quantizer) and the bound will only be a function of the corresponding source distortion.*

We next show how the results obtained up to this point can be used to devise a framework for analyzing the tradeoff between latency and distortion. In Lemma 5, we showed that the overall expected distortion $\mathbb{E}[D_{\text{TV}}(\mathbf{p}, \hat{\mathbf{p}}(\kappa(\mathbf{y}))]$ can be upper bounded by $(1-\epsilon^*(n))\beta_s + \epsilon^*(n)$. For brevity, we refer to $\mathbb{E}[D_{\text{TV}}(\mathbf{p}, \hat{\mathbf{p}}(\kappa(\mathbf{y}))]$ as β_t (i.e., the total expected distortion). To derive a relationship between the overall distortion β_t and latency T, we first recall the finite blocklength result from [8], which states that the decoding error probability $\epsilon^*(n)$ that can be assured for sending J bits through an AWGN channel is given by [8] [44]:

$$\epsilon^*(n, \gamma, J) = Q\left(\frac{nC(\gamma) - J + \frac{1}{2}\log_2 n}{\sqrt{nV(\gamma)}}\right), \tag{13}$$

where n represents the blocklength, γ represents the SNR, $C(\gamma)$ represents the capacity defined by $\frac{1}{2}\log_2(1+\gamma)$ and $V(\gamma)$ denotes the channel dispersion defined by $\frac{\gamma(\gamma+2)}{2(\gamma+1)^2}(\log_2(e))^2$.

It should be noted that (13) is derived assuming that the messages being sent are equally likely to be chosen. However, the results generated by a classifier (i.e. probability vector) are entirely dependent on the data it is given (i.e. image, time series, etc.) As such, we do not have any prior knowledge regarding the underlying distribution of the data. Thus, to utilize the results from the finite blocklength literature, we assume that all probability vectors are equally likely. We also note that making this assumption does not prevent using the derived results on real-world data. As Section IV-C indicates, our obtained experimental results corroborate the insights derived in this paper and help validate the utility of our framework.

Let us now return to the problem of transmitting a probability vector $\mathbf{p}$ over an AWGN channel. Observe that the number of bits one can use to represent $\mathbf{p}$ can be chosen as a function of the source distortion β_s (via Lemmas 2-4, i.e., $J(\beta_s)$). However, the choice of β_s, and therefore $J(\beta_s)$ also directly impact the decoding error probability $\epsilon^*(n, \gamma, J(\beta_s))$ as given in (13). Thus, the resulting overall distortion from Lemma 5 can then be bounded by $(1-\epsilon^*(n, \gamma, J(\beta_s)))\beta_s + \epsilon^*(n, \gamma, J(\beta_s))$. Hence, if we are given a target total expected distortion of β_t, one can then optimize β_s to minimize latency while satisfying the total distortion budget. This is the core idea behind our approach and is formalized in the following Theorem.

Theorem 1. *Given a total distortion budget β_t, for a certain quantization technique we can achieve the following latency*

assuming an AWGN channel:

$$T(\beta_t) = \min_{0 \le \beta_s \le \beta_t} \frac{n(\beta_s)}{2B} \tag{14}$$

where

$$\sqrt{n(\beta_s)} = \frac{r + \sqrt{r^2 + 4C(\gamma)J(\beta_s)}}{2C(\gamma)}, \tag{15}$$

and $r = \sqrt{V(\gamma)}Q^{-1}\left(\frac{\beta_t - \beta_s}{1-\beta_s}\right)$.

Proof. Our objective is to minimize latency while satisfying a constraint on the overall distortion β_t. First, the number of bits to quantize $\mathbf{p}$ can be obtained based on the choice of quantization scheme (e.g. $J(\beta_s) = 2k\log_2\left(\frac{k}{\beta_s}\right)$ for UQ on Lemma 2). We can then rearrange Lemma 5 to solve for the desired block error probability $\epsilon^*(n, \gamma, J(\beta_s)) = \frac{\beta_t - \beta_s}{1-\beta_s}$ in terms of β_t & β_s. Hence, the next step is to find the minimum number of channel uses (n) that can support the desired block error probability of $\frac{\beta_t - \beta_s}{1-\beta_s}$ by using (13). Specifically, we wish to solve for the smallest non-negative integer n satisfying:

$$\left(\frac{\beta_t - \beta_s}{1-\beta_s}\right) \le Q\left(\frac{nC(\gamma) - J(\beta_s) + \frac{1}{2}\log_2 n}{\sqrt{nV(\gamma)}}\right). \tag{16}$$

As the $Q(\cdot)$ function (complementary CDF of standard Gaussian) is monotonically decreasing, this means that for any $n > 1$, we can bound the r.h.s. of (16) as:

$$Q\left(\frac{nC(\gamma) - J(\beta_s) + \frac{1}{2}\log_2 n}{\sqrt{nV(\gamma)}}\right) \le Q\left(\frac{nC(\gamma) - J(\beta_s)}{\sqrt{nV(\gamma)}}\right). \tag{17}$$

Thus, we can find n by instead solving for the simpler equation $\left(\frac{\beta_t - \beta_s}{1-\beta_s}\right) = Q\left(\frac{nC(\gamma) - J(\beta_s)}{\sqrt{nV(\gamma)}}\right)$. Applying $Q^{-1}(\cdot)$ on both sides, we arrive at the following:

$$nC(\gamma) - \sqrt{nV(\gamma)}Q^{-1}((\beta_t - \beta_s)/(1-\beta_s)) - J(\beta_s) = 0. \tag{18}$$

This equation can be viewed as a quadratic by setting $\tilde{n} = \sqrt{n}$. Solving for n, we arrive at the latency expression (setting $T = n/2B$). One can then optimize the latency by minimizing over all $\beta_s \in [0, \beta_t]$, thus completing the proof of Theorem 1. □

C. Generalization to Fading Channels

In this section, we now extend our framework to derive results for fading channels. To accomplish this, we leverage finite blocklength results for coherent and non-coherent fading channels [31]. The probability of error $\epsilon^*(n)$ for sending $J(\beta_s)$ bits through a Rayleigh fading channel assuming access to channel state information (CSI) at the receiver is given by [31]:

$$\epsilon^*(n, \gamma, J(\beta_s)) = Q\left(\frac{nC_c(\gamma) - J(\beta_s)\ln(2)}{\sqrt{nFV_c(F, \gamma)}}\right), \tag{19}$$

where F represents the coherence interval, C_c is the capacity defined as $E[\log(1 + \gamma Z_1)]$ (where Z_1 is a sequence of variables samples from the Gamma(1,1) distribution), and $V_c(F, \gamma)$ is the channel dispersion given as $var[\log(1+\gamma Z_1)] + \frac{1}{F} - \frac{1}{F}E[\frac{1}{1+\gamma Z_1}]^2$.

Similarly, an approximation for the error probability sending $J(\beta_s)$ bits through a Rayleigh fading channel without CSI was derived in [31] for high SNRs assuming that $0 < \epsilon^*(n) < \frac{1}{2}$. The approximation is as follows:

$$\epsilon^*(n, \gamma, J(\beta_s)) = Q\left(\frac{n\underline{I}(F, \gamma) - J(\beta_s)F\ln(2)}{\sqrt{nF\tilde{U}(F)}}\right), \tag{20}$$

where $\underline{I}(F, \gamma)$ can be approximated as $(F-1)\log(F\gamma) - \log\Gamma(F) - (F-1)(1+\eta) + K_I'(F, \gamma)$, η represents Euler's constant, Γ is the Gamma function, $\tilde{U}(F) = (F-1)^2\frac{\pi^2}{6} + (F-1)$, and K_I' is a function that must be 0 as $\gamma \to \infty$ and $F > 2$. In this work, we assume $K_I'(F, \gamma) = \frac{F}{5\gamma}$.

With respect to (19) & (20), recall that in this work, J is determined based on a given requirement on the source distortion β_s and choice of quantization: UQ, LO, or SLQ. Theorem 1 introduced our framework for finding the optimal β_s to achieve the minimum latency for a specific total distortion β_t. Following similar steps as shown in the proof for Theorem 1, and using (19) & (20), the following theorems can be obtained for analyzing the latency-distortion tradeoff in fading channels with/without CSI.

Theorem 2. *Given a total distortion budget β_t, for a certain quantization technique we can achieve the following latency assuming a Rayleigh fading channel with CSI at the receiver:*

$$T(\beta_t) = \min_{0 \le \beta_s \le \beta_t} \frac{n(\beta_s)}{2B} \tag{21}$$

where

$$\sqrt{n(\beta_s)} = \frac{r + \sqrt{r^2 + 4C_c(\gamma)J(\beta_s)\ln 2}}{2C(\gamma)}, \tag{22}$$

and $r = \sqrt{FV_c}Q^{-1}\left(\frac{\beta_t - \beta_s}{1-\beta_s}\right)$.

Theorem 3. *Given a total distortion budget β_t, for a certain quantization technique we can achieve the following latency assuming a Rayleigh fading channel at high SNRs without CSI:*

$$T(\beta_t) = \min_{0 \le \beta_s \le \beta_t} \frac{n(\beta_s)}{2B} \tag{23}$$

where

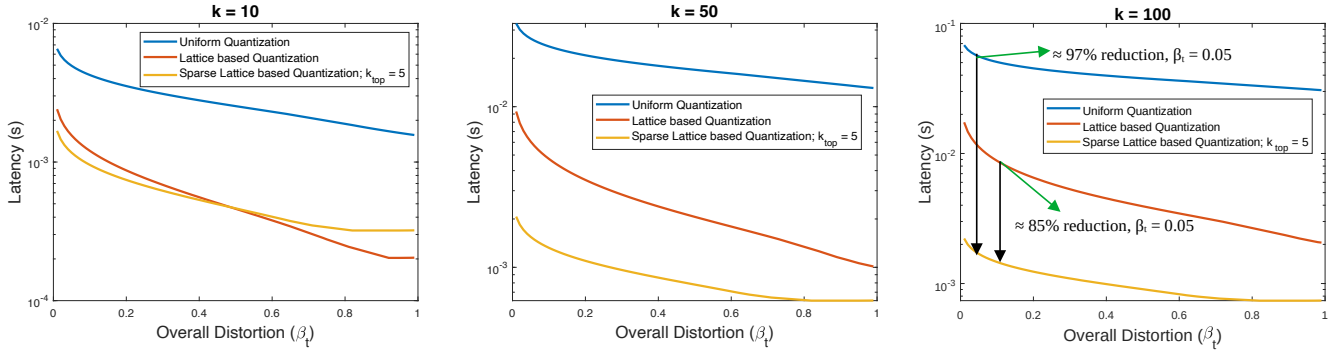
$$\sqrt{n(\beta_s)} = \frac{r + \sqrt{r^2 + 4\underline{I}(F, \gamma)J(\beta_s)F\ln 2}}{2C(\gamma)}, \tag{24}$$

and $r = \sqrt{F\tilde{U}(F)}Q^{-1}\left(\frac{\beta_t - \beta_s}{1-\beta_s}\right)$, *with* $0 < \frac{\beta_t - \beta_s}{1-\beta_s} < \frac{1}{2}$.

IV. Experimental Results

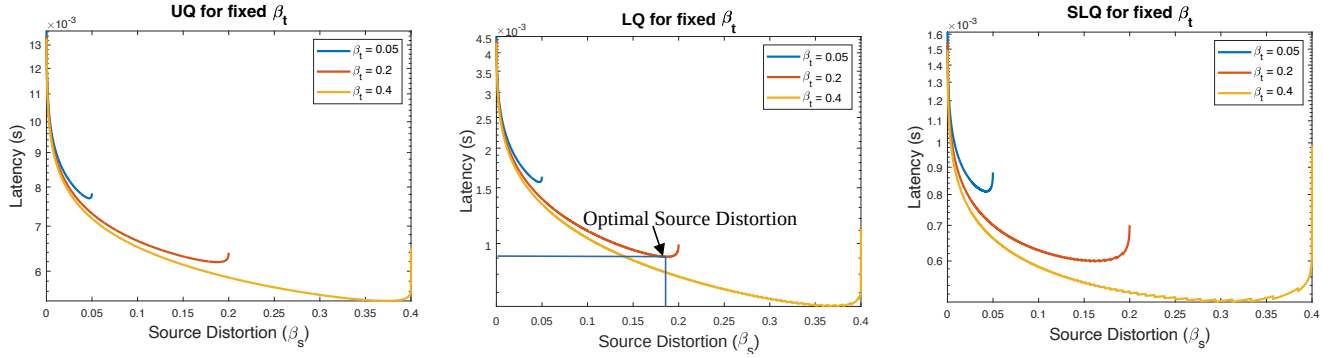
In this section, we present results which illustrate the tradeoff between the latency and overall distortion for sending a probability vector to a receiver over AWGN and fading channels for UQ, LQ, and SLQ. Unless otherwise stated, we assume that $B_0 = 10$ kHz, $\delta = 0.00001$ for SLQ, and $0 < \epsilon^*(n) < 0.5$.

This article has been accepted for publication in IEEE Transactions on Communications. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2024.3484939



(a) (b) (c)

Fig. 4. Lower convex hull of latencies for different β_t for UQ, LQ, and SLQ (obtained from Theorem 1). Results are reported for $k = 10(a), 50(b)$ & $100(c)$ to observe the impact varying the number of classes has on the quantization schemes.



(a) (b) (c)

Fig. 5. Impact of varying source distortion on incurring latency for fixed $\beta_t = 0.05, 0.2$ & 0.4. Collecting results for (a) UQ, (b) LQ, (c) and SLQ assuming $k = 70$ & $k_{top} = 20$. It can be observed that as β_t increases, the source distortion that obtains the minimal latency also increases. It can also be observed that SLQ is able to obtain the lowest latencies out of the three techniques.

A. Comparison of Quantization techniques

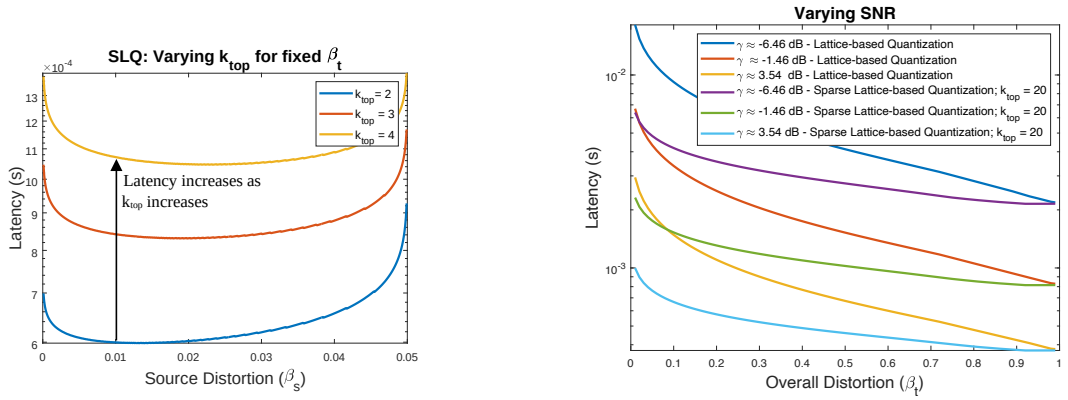
We first observe the trade-off between latency and distortion for the uniform and lattice-based quantization techniques for the AWGN channel. In this section, we use the theoretical results from Lemmas 2-4 and Theorem 1 to determine the latencies for each quantization technique.

Minimum latency for a fixed β_t & different k: Figure 4 reports results comparing the incurred latencies for the three quantization methods by solving the optimization problem in Theorem 1 as the number of classes is varied for $k = 10, 50$ & 100. We assume that $B = 320$ kHz and $\gamma_0 = 5$ dB, which yields an SNR of $\gamma \approx -10.1$ dB. We also assume $k_{\text{top}} = 5$ for SLQ. The figure indicates that LQ and SLQ can incur lower latencies over an AWGN channel compared to UQ as the number of classes is varied. Additionally, the figure indicates that as k increases, the latency reduction from LQ to SLQ increases. At $k = 100$ and $\beta_t = 0.05$, for example, SLQ can attain a latency reduction of approximately 97% and 85% with respect to UQ and LQ. However, it's interesting to note that for low k and high β_t, Figure 4a indicates that LQ can incur less latencies compared to SLQ. This emphasizes that when using SLQ, more benefits are obtained at higher k.

Interplay between source distortion (β_s) and Latency: Fig. 5 compares the quantization schemes for a fixed total distortion β_t with $k = 70$ classes and $k_{top} = 20$ for SLQ. We also set $B = 100$ kHz and $\gamma_0 = 15$ dB, which results in $\gamma = 5$ dB. Each of the figures indicate, that for higher dimensional probability vectors, a reduction in latency can be achieved when more source distortion is allowed. However, as the total distortion is increased, the optimal source distortion increases for each of the lattice-based methods. The figure also indicates that the lattice based quantizers can attain lower latencies compared to UQ, with SLQ performing better than LQ when quantizing high dimensional probability vectors. Each of the figures also indicate that surges in the latency occur as β_s approaches β_t. This intuitively makes sense because as β_s approaches β_t, this implies that no compensation for the distortion is being performed by the source encoder/decoder. This means that the channel encoder/decoder are responsible for satisfying the requirement on β_t and can only do so by using higher blocklengths.

Impact of degree of sparseness: Figure 6a presents the effect of varying k_{top} (i.e. how many of the highest probabilities are chosen for transmission) has on the latency for a fixed β_t when using SLQ. It's assumed that $B = 100$ kHz and $\gamma_0 = 8$ dB resulting in $\gamma = -2$ dB. Similar to Figure 4a, to observe

This article has been accepted for publication in IEEE Transactions on Communications. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCOMM.2024.3484939



(a) (b)

Fig. 6. (a) Impact of varying k_{top} for sparse lattice-based quantization on latency as a function of source distortion, with $k = 1000$ classes and $\beta_t = 0.05$. As expected, a lower choice of k_{top} incurs lower latency. (b) Lower convex hull of observed latencies as a function of β_{t} for different SNRs ($-6.46, -1.46, 3.54$ dB) with $k = 100$ classes.

the full merit of the method, $k = 1000$. The figure indicates that as k_{top} increases, the latency also increases, which is as expected as this means more predictions are included in the sparse vector. The figure also indicates, similar to Figure 5, that as more source distortion is allowed, smaller latencies can be attained. This means that to quantize and send additional values from the probability vector at a lower latency, more source distortion must be allowed.

Latency as a function of SNR: Figure 6b presents the latencies incurred for LQ and SLQ at different SNRs. Recall that the SNR is related to the bandwidth B as $\gamma = \frac{\gamma_0 B_0}{B}$; thus by varying the reference SNR γ_0, different SNRs γ can be simulated. In Figure 6b, γ_0 is varied to 5 dB, 10 dB, and 15 dB respectively, which corresponds to γ of $-6.46, -1.46, 3.54$ dB respectively. To observe the full benefits of SLQ, $k = 100$ and $k_{\text{top}} = 20$. The figure indicates that for both techniques, the incurred latency decreases as the SNR increases. Figure 6b also indicates that SLQ significantly outperforms LQ at each of the simulated SNRs. This means that for a high k-dimensional probability vector, SLQ can incur lower latencies even in poor channel conditions.

B. Application to Fading Channels

We now analyze our framework considering Rayleigh block-fading channels using the finite blocklength approximations presented in Section III-C. In this section, we use the theoretical results from Lemmas 2-4 and Theorems 2-3 to determine the latencies for each quantization technique. We assume $k_{top} = 16$, $B = 100$ kHz, $\gamma_0 = 11$ dB, and $k = 100$, which results in $\gamma = 1$ dB. Figure 7a shows the latency-distance tradeoff for the three quantization schemes for a Rayleigh fading channel assuming CSI at the receiver and $F = 20$. The figure indicates that, similar to our previously presented results, SLQ can still outperform UQ and LQ. Figure 7b shows the latency-distortion tradeoff for the three quantization schemes for a Rayleigh fading channel without CSI. As (20) is a high SNR approximation, we have adjusted the following parameters: $B = 200kHz$, $B_0 = 800kHz$, $\gamma_0 = 15$ dB, which results in $\gamma \approx 21$ dB. Additionally, k is set to 1000 classes, with $k_{\text{top}} = 70$ for SLQ. It can be observed that even without access to CSI, similar trends are observed with SLQ performing significantly better than UQ & LQ.

C. Experimental Validation & Accuracy vs. Blocklength Tradeoffs

In this section, we study the relationship between blocklengths required by the quantization techniques and the achieved accuracies to help navigate the latency-accuracy tradeoff. Predictions from real datasets are quantized, coded with a $\frac{1}{2}$ rate convolutional code, modulated using baseband BPSK and sent through a simulated channel. After imposing channel effects, the signal is demodulated, decoded, and unquantized and the class with the highest probability is identified. We use the metric 'relative accuracy' to measure how often the class assigned with the highest probability by the classifier at the transmitter is correctly identified after transmission. For each technique, we optimize the source distortion that enables the smallest blocklength needed to achieve a certain relative accuracy and use Lemmas $2-4$ to calculate the respective bit budgets. In these experiments, for SLQ, we use k bits to represent the indices of the k_{top} highest probabilities (recall that $\left\lceil \log_2 \binom{k}{k_{\text{top}}} \right\rceil$ serves as a lower bound). Tables I & II shows results for predictions made on the CIFAR-10, CIFAR-100 and UCF-101 datasets using a convolutional neural network [45], VGG architecture [32] [33] and I3D network [38] [39] respectively. Each reported relative accuracy is averaged over 7 passes of 9500 predictions made on CIFAR-10 and 10000 predictions made on CIFAR-100 & UCF-101. Tables I & II show the blocklengths needed to achieve relative accuracies greater than 95% and 89% respectively for each of the quantization techniques assuming SNRs of approximately 5 & 14 dB. It should be noted that for SLQ, the dimension of the lattice used may vary based on the mass of the non k_{top} values in each set of prediction results; recall from Theorem 3 that $\ell = \left\lceil \frac{k_{\text{top}}}{4(\beta_s - \delta)} \right\rceil$. To account for this, the table presents

the mean and standard deviation of the blocklength used for SLQ, where $k_{top} = 4$ for the predictions made on CIFAR-10 dataset and $k_{top} = 13$ for the predictions made on the CIFAR-100/UCF-101 datasets.

Table I indicates that for CIFAR-10, to achieve a relative accuracy greater than 95%, the blocklengths needed for LQ and SLQ are significantly smaller than the blocklength needed by UQ at both 5 & 14 dB. This implies that LQ and SLQ can incur smaller latencies than UQ while also performing comparably to it. For CIFAR-100, a similar trend is observed where the blocklength needed for SLQ to achieve a relative accuracy greater than 95% is significantly smaller than the blocklength needed for UQ at both SNRs. This trend can also be observed on the UCF-101 dataset at 14 dB, where UQ and SLQ are being used on predictions made over 400 classes. At 5 dB, however, both UQ and SLQ are not able to achieve greater than 95% relative accuracy for the UCF-101 dataset. Table II shows the blocklengths needed for UQ, LQ, and SLQ to achieve a relative accuracy greater than 89% on the same sets of predictions. As in Table I, LQ and SLQ require significantly lower blocklengths than UQ (and subsequently lower transmission latencies) but can still provide comparable performance to UQ. Additionally, the table indicates that as the threshold for relative accuracy is lowered (from 95% with respect to Table I), there can be additional reductions in the required blocklength for lattice-based techniques. Looking at CIFAR-10 at $\gamma \approx 14$ dB, for example, the blocklengths needed for LQ to achieve a relative accuracy greater than 95% and 89% are 22 and 16 respectively. Thus, Tables I & II indicate that LQ and SLQ while incurring smaller blocklengths (and subsequently smaller transmission latencies) than UQ can still perform comparably to UQ.

Dataset	Technique	$\gamma \approx 14$ dB	$\gamma \approx 5$ dB
CIFAR-10	UQ	120	120
	LQ	**22**	**22**
	SLQ	30.14 ± 0.58	30.21 ± 0.71
CIFAR-100	UQ	2600	2600
	SLQ	**222**	**222**
UCF-101	UQ	13600	n/a
	SLQ	$\mathbf{828.13 \pm 4.03}$	n/a

TABLE I
BLOCKLENGTHS TO ACHIEVE RELATIVE ACCURACIES $> 95\%$ OVER AWGN CHANNEL FOR UQ, LQ, AND SLQ ON PREDICTIONS MADE ON CIFAR-10, CIFAR-100 AND UCF-101.

Dataset	Technique	$\gamma \approx 14$ dB	$\gamma \approx 5$ dB
CIFAR-10	UQ	120	120
	LQ	**16**	**16**
	SLQ	29.71 ± 0.77	29.71 ± 0.77
CIFAR-100	UQ	2600	2600
	SLQ	**222**	**222**
UCF-101	UQ	13600	13600
	SLQ	$\mathbf{822.36 \pm 1.43}$	$\mathbf{830.63 \pm 3.31}$

TABLE II
BLOCKLENGTHS TO ACHIEVE RELATIVE ACCURACIES $> 89\%$ OVER AWGN CHANNEL FOR UQ, LQ, AND SLQ ON PREDICTIONS MADE ON CIFAR-10, CIFAR-100 AND UCF-101.

To further investigate the latency-accuracy tradeoff, we analyze the full end-to-end delay (encoding, transmission, and decoding latencies) when using UQ, LQ, and SLQ for an AWGN channel. Table III reports the end-to-end delays for UQ, LQ and SLQ on predictions made on CIFAR-10 and UQ/SLQ for predictions made on CIFAR-100. The average end-to-end delay was calculated for 4 separate passes through the predictions on each dataset, with Table III presenting the average across the four runs. We use the same setup used to generate the results in Table I, including the same source distortions for each technique to enable the smallest blocklength that still reaches a 95% relative accuracy for $\gamma \approx 14$ dB for each dataset. We calculate the transmission latency using (2) assuming a bandwidth of 100 kHz. For CIFAR-10, similar to Table I, the results show that the lattice-based techniques incur lower end-to-end latencies than UQ while still performing comparably to UQ. Similarly, for CIFAR-100, the table also indicates that SLQ incurs a lower end-to-end latency than UQ while still satisfying the 95% relative accuracy threshold.

Technique	Dataset	Average End-to-End Delay (ms)
UQ	CIFAR-10	0.95
LQ	CIFAR-10	0.35
SLQ ($k_{\text{top}} = 4$)	CIFAR-10	0.48
UQ	CIFAR-100	15.4
SLQ ($k_{\text{top}} = 13$)	CIFAR-100	1.5

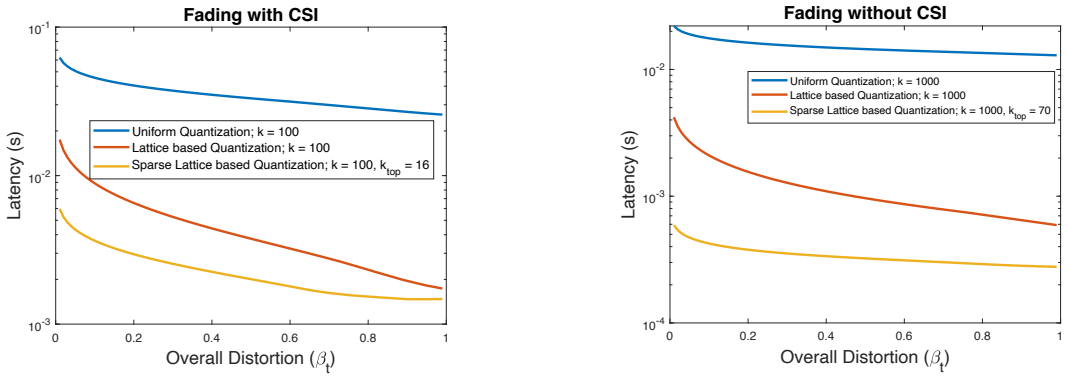
TABLE III
AVERAGE END-TO-END LATENCIES (IN MS) OVER 4 RUNS OF USING UQ, LQ AND SLQ ON PREDICTIONS MADE ON CIFAR-10 AND UQ/SLQ ON PREDICTIONS MADE ON CIFAR-100.

D. Collaborative Reasoning via Classifier Predictions

In this section, we investigate a collaborative scenario where L transmitters pass their noisy observation of an input x to their respective local classifiers. The results (i.e. probability vectors) are quantized and transmitted through multiple channels of varying quality. The receiver must deduce which class the classifier would have assigned the highest probability if it were provided a noiseless version of x by leveraging information from the received distorted probability vectors. This is akin to the work in [46], where two transmitters monitoring the same input pass images through a deep learning based joint source channel coding method to send the most representative features to a receiver which subsequently performs image retrieval. Two strategies are used in this section:

- *Majority voting:* After decoding the received probability vectors, the highest class from each vector is chosen. Whichever class is chosen the most frequent from the vectors, is used as the final answer.
- *Averaging:* Average the received decoded vectors and choose the class with the highest probability from this averaged vector.

Table IV shows the average relative accuracy for predictions made on the CIFAR-10 dataset and sent through an AWGN channel using UQ and LQ. We assume the same setup as in Section IV-C, with a $\frac{1}{2}$ rate convolutional code and baseband BPSK modulation. In this setup, we assume that each transmitter has the same ML classifier. The source distortion is set to 0.35 for all reported results; meaning that UQ and LQ require blocklengths of 180 and 30 respectively. Each point in the table is the average relative accuracy after taking 7 passes through a set of 9500 predictions. An important note to make is that under both majority voting and averaging, LQ with only a $\frac{1}{6}^{th}$ of the blocklength of UQ is able to perform comparably to UQ. For three AWGN channels with $\gamma \approx (-2, -1, 0)$ dB, the table



(a) (b)

Fig. 7. (a) Lower convex hull of latencies for different β_t for UQ, LQ, and SLQ for a Rayleigh fading channel assuming CSI at the receiver. (b) Lower convex hull of latencies for different β_t for UQ, LQ, and SLQ for a Rayleigh fading channel without CSI at the receiver.

indicates that when a limited number of channels are available, and of extremely poor quality, the averaging method performs much better than using a majority vote. For channels with $\gamma \approx (1, 2, 3)$ dB, the table indicates that when the channels are of poor but similar quality, the improvement gained by averaging the received vectors over using a majority vote shrinks significantly. However, when the number of available channels is increased to seven, each of relatively poor quality, the table indicates that using a majority vote gives a very slight improvement in performance compared to averaging. This implies that for a limited and larger number of channels, majority voting and averaging should be used respectively. Other fusion strategies could also be employed in this setup, which is left as future work.

Num. of Channels	Approx. SNR set (dB)	Technique	Majority vote Acc.	Averaging Acc.
3	(-2, -1, -0.01)	UQ	59.15%	**80.43%**
		LQ	68.47%	**79.47%**
3	(1,2,3)	UQ	97.82%	**98.70%**
		LQ	94.72%	**95.43%**
3	(3,5,8)	UQ	**99.94%**	99.75%
		LQ	97.39%	**97.44%**
7	(-2,-1,-0.01,1,2,3,4)	UQ	**99.12%**	98.48%
		LQ	**96.84%**	96.31%

TABLE IV
RELATIVE ACCURACIES FOR TRANSMITTING CIFAR-10 PREDICTIONS ACROSS MULTIPLE AWGN CHANNELS WITH UQ & LQ.

V. CONCLUSION

In this work, we have investigated a framework where the decisions (a probability vector) from a classification task are transmitted over a noisy channel. Specifically, we study the tradeoff between the latency associated with transmitting this result against the distortion incurred with quantizing the result and the impact of channel noise on the transmission. To accomplish this, we have analyzed the performance of uniform and lattice-based quantization techniques by first providing results bounding the necessary bit budgets under each technique to satisfy a requirement on the allowable source distortion. Then by linking distortion due to decoding errors (using results from finite blocklength channel capacity) with the distortion due to quantization, we are able to create a framework that allows us to find an optimized source distortion that achieves a minimal transmission latency at different levels of end-to-end distortion. Our results show that there is an interesting interplay between source distortion (i.e., distortion for the probability vector measured via f-divergence) and the subsequent channel encoding/decoding parameters; and indicate that a *joint* design of these parameters is crucial to navigate the latency-distortion tradeoff. After varying different parameters of the framework, and assuming both AWGN and fading channels, our results show that sparse-lattice based quantization is the most efficient at minimizing latency at different levels of end-to-end distortion. Specifically, our results indicate that sparse-lattice based quantization outperforms all other methods for high dimensional probability vectors (i.e. a higher number of classes) and sparse predictions generated by the classifier (which is often the case in various ML classifiers, as also evidenced in CIFAR-100, Imagenet-1K, and Kinetics-400 datasets). We believe that the sparse lattice based quantization techniques could also be useful for other ML based systems requiring low latency, such as in transmitting semantic information.

APPENDIX
PROOF OF LEMMA 1

The minimal achievable latency $T^*(\beta)$ is a non-increasing function of β. This is clear from the fact that any decoder which satisfies a distortion constraint of β also satisfies the distortion constraint of β' for $\beta' \geq \beta$.

We next show that $T^*(\beta)$ is a convex function of β. Let $T^*(\beta_{t_1})$ ($T^*(\beta_{t_2})$, respectively) represent the minimum latencies obtained using encoder-decoder pair $(\mathcal{E}_1^*, \mathcal{D}_1^*)$ ($(\mathcal{E}_2^*, \mathcal{D}_2^*)$, respectively) that satisfy $D_{\mathrm{f}}(\mathbf{p}, \hat{\mathbf{p}}_1) \leq \beta_{t_1}$ ($D_{\mathrm{f}}(\mathbf{p}, \hat{\mathbf{p}}_2) \leq \beta_{t_2}$, respectively). We define a new encoder-decoder pair $(\mathcal{E}_3, \mathcal{D}_3)$ such that,

$$(\mathcal{E}_3, \mathcal{D}_3) = \begin{cases} (\mathcal{E}_1^*, \mathcal{D}_1^*) & \text{with probability } \alpha \\ (\mathcal{E}_2^*, \mathcal{D}_2^*) & \text{with probability } 1-\alpha. \end{cases}$$

The expected latency using $(\mathcal{E}_3, \mathcal{D}_3)$ is $\alpha T^*(\beta_{t_1}) + (1-\alpha)T^*(\beta_{t_2})$. The total distortion using $(\mathcal{E}_3, \mathcal{D}_3)$ is $D_{\mathrm{f}}(\mathbf{p}, \alpha\hat{\mathbf{p}}_1 +$

$(1-\alpha)\hat{\mathbf{p}}_2)$, which can be upper bounded as,

$$D_{\mathrm{f}}(\mathbf{p}, \alpha\hat{\mathbf{p}}_1 + (1-\alpha)\hat{\mathbf{p}}_2) \overset{(a)}{\leq} \alpha D_{\mathrm{f}}(\mathbf{p}, \hat{\mathbf{p}}_1) + (1-\alpha) D_{\mathrm{f}}(\mathbf{p}, \hat{\mathbf{p}}_2) \overset{(b)}{\leq} \alpha\beta_{t_1} + (1-\alpha)\beta_{t_2}, \tag{25}$$

where (a) follows from convexity of f-divergence, and (b) follows from the bounds on end-to-end distortion for the two individual decoders. Let us now consider the optimal encoder-decoder pair $(\mathcal{E}^*, \mathcal{D}^*)$ that satisfies the distortion constraint $D_{\mathrm{f}}(\mathbf{p}, \hat{\mathbf{p}}) \leq \alpha\beta_{t_1} + (1-\alpha)\beta_{t_2}$. The minimum latency using $(\mathcal{E}^*, \mathcal{D}^*)$ is then $T^*(\alpha\beta_{t_1} + (1-\alpha)\beta_{t_2})$. Recall, $(\mathcal{E}^*, \mathcal{D}^*)$ is optimal encoder-decoder pair, and therefore, the corresponding latency must be always less than the latencies obtained using any $(\mathcal{E}_3, \mathcal{D}_3)$ pair. That is,

$$T^*(\alpha\beta_{t_1} + (1-\alpha)\beta_{t_2}) \leq \alpha T^*(\beta_{t_1}) + (1-\alpha) T^*(\beta_{t_2}).$$

This proves that $T^*(\beta)$ is convex. $D^*(T)$ can be shown to be convex in a similar manner by leveraging the convexity of f-divergence.

Proof of Lemma 2

We consider uniform quantization (UQ) with bins of width $\frac{1}{2^j}$, where $j = \lfloor J/k \rfloor$. For $r \in \{1, 2, \cdots, 2^j - 1\}$, we define the values for the r^{th} bin in the range $\left[\frac{r}{2^j}, \frac{r+1}{2^j}\right)$. In other words, for any $i \in [k]$, $\mathbf{q}[i]$ is obtained by mapping the value of $\mathbf{p}[i]$ in the range $\left[\frac{r_i}{2^j}, \frac{r_i+1}{2^j}\right)$ to $\frac{r_i+\frac{1}{2}}{2^j}$. However, we note that $\mathbf{q}$ may not necessarily be a probability vector. We define the probability vector $\tilde{\mathbf{q}}$, by normalizing the values in $\mathbf{q}$. Therefore, for any $i \in [k]$, we can write $\mathbf{p}[i]$ and $\tilde{\mathbf{q}}[i]$ as follows:

$$\mathbf{p}[i] = \frac{r_i + \delta_i}{2^j}, \qquad \tilde{\mathbf{q}}[i] = \frac{r_i + \frac{1}{2}}{S \cdot 2^j}, \tag{26}$$

where $\delta_i \in [0, 1)$ and $S = \sum_{i=1}^{k} \frac{r_i+\frac{1}{2}}{2^j}$. Returning to our goal, recall that we wish to pick J such that $D_{\mathrm{TV}}(\mathbf{p}, \tilde{\mathbf{q}}) \leq \beta_s$. To this end, we first bound

$$\begin{aligned} D_{\mathrm{TV}}(\mathbf{p}, \tilde{\mathbf{q}}) &= \frac{1}{2}\sum_{i=1}^{k} |\mathbf{p}[i] - \tilde{\mathbf{q}}[i]| \\ &= \frac{1}{2}\sum_{i=1}^{k} \left| \frac{r_i + \delta_i}{2^j} - \frac{r_i + \frac{1}{2}}{S \cdot 2^j} \right| \\ &= \frac{1}{2^{(j+1)}}\sum_{i=1}^{k} \left| r_i\left(1 - \frac{1}{S}\right) + \delta_i - \frac{1}{2S} \right| \\ &\overset{(a)}{\leq} \frac{1}{2^{(j+1)}}\sum_{i=1}^{k} \left(\left| r_i\left(1 - \frac{1}{S}\right) \right| + \left| \delta_i - \frac{1}{2S} \right| \right) \\ &\overset{(b)}{\leq} \frac{1}{2^{(j+1)}}\sum_{i=1}^{k} \left(\left| 2^j\left(1 - \frac{1}{S}\right) \right| + \left| \delta_i - \frac{1}{2S} \right| \right), \end{aligned} \tag{27}$$

where (a) follows from triangle inequality, and (b) follows from the fact that $r_i < 2^j$. We also know that, $\sum_i^k \mathbf{p}[i] = 1$ and $\sum_i^k \tilde{\mathbf{q}}[i] = S$. Consider the difference,

$$\left| \sum_{i=1}^{k} \frac{r_i + \delta_i}{2^j} - \sum_{i=1}^{k} \frac{r_i + \frac{1}{2}}{2^j} \right| \leq \sum_{i=1}^{k} \frac{|\delta_i - \frac{1}{2}|}{2^j} \leq \frac{k}{2^{j+1}}. \tag{28}$$

Therefore, we have that,

$$|1 - S| \leq \frac{k}{2^{j+1}}. \tag{29}$$

Suppose that j is given as

$$j = \log_2\left(\frac{k}{2\alpha}\right) \tag{30}$$

for some $\alpha \in (0, 0.5]$. Therefore, from (29) we have that,

$$1 - \alpha \leq S \leq 1 + \alpha. \tag{31}$$

Using (30) and (31) in (27), we can further bound $D_{\mathrm{TV}}(\mathbf{p}, \tilde{\mathbf{q}})$ as,

$$\begin{aligned} D_{\mathrm{TV}}(\mathbf{p}, \tilde{\mathbf{q}}) &\leq \left[\frac{\alpha k}{2(1-\alpha)} + \max\left\{ \frac{\alpha}{2(1-\alpha)}, \alpha - \frac{\alpha}{2(1+\alpha)} \right\} \right] \\ &\overset{(a)}{\leq} \frac{\alpha k}{2(1-\alpha)} + \frac{\alpha}{2(1-\alpha)} = \frac{\alpha}{1-\alpha}\left(\frac{k+1}{2}\right), \end{aligned} \tag{32}$$

where (a) holds for all $\alpha \in (0, 0.5]$. Now, since we require $D_{\mathrm{TV}}(\mathbf{p}, \tilde{\mathbf{q}}) \leq \beta_s$, we can pick α such that $\frac{\alpha}{1-\alpha}\left(\frac{k+1}{2}\right) \leq \beta_s$. We can pick α to satisfy this constraint with equality, i.e.,

$$\alpha^* = \frac{2\beta_{\mathrm{s}}}{k + 1 + \beta_{\mathrm{s}}}. \tag{33}$$

Next, substituting (33) in (30), we can then claim that as long as the total bit budget $J = kj \geq k\log_2(k/2\alpha^*)$, then $D_{\mathrm{TV}}(\mathbf{p}, \tilde{\mathbf{q}}) \leq \beta_s$. Now, using the fact that $k \geq 2$ and $\beta_s \in [0, 1]$, it can be readily verified that $2k \cdot \log_2\left(\frac{k}{\beta_s}\right) \geq k\log_2(k/2\alpha^*)$, completing the proof of Lemma 2.

Proof of Lemma 3

We denote the probability distribution on Q_ℓ closest to a given probability vector $\mathbf{p}$ as $\mathbf{q}_{\mathrm{LQ}}(\mathbf{p})$. Recall from [19] that by performing lattice-based quantization (LQ), the distortion incurred is given by $D(\mathbf{p}, \mathbf{q}_{\mathrm{LQ}}(\mathbf{p})) = \frac{k}{4\ell}$. For our framework, we propose setting $\ell = \left\lceil \frac{k}{4\beta_s} \right\rceil$. In doing so, we can obtain the following upper bound on $D(\mathbf{p}, \mathbf{q}_{\mathrm{LQ}}(\mathbf{p}))$:

$$\begin{aligned} D(\mathbf{p}, \mathbf{q}_{\mathrm{LQ}}(\mathbf{p})) &= \frac{k}{4\ell} \\ &= \frac{k}{4\left\lceil \frac{k}{4\beta_s} \right\rceil} \\ &\leq \frac{k}{4\left(\frac{k}{4\beta_s}\right)} \\ &= \beta_s. \end{aligned}$$

Thus, sending $\left\lceil \log_2\binom{\ell+k-1}{k-1} \right\rceil$ bits under LQ, with $\ell = \left\lceil \frac{k}{4\beta_s} \right\rceil$ will satisfy the source distortion requirement.

Proof of Lemma 4

For sparse lattice-based quantization (SLQ), we assume $\mathbf{p}$ has the following property: $\sum_{i \in k_{\text{top}}} \mathbf{p}[i] \geq 1 - \delta$, where $\delta \in [0, 1]$. In other words, we assume that the k_{top} highest values constitute a significant portion of the mass of $\mathbf{p}$. This implies that $\sum_{i \notin k_{\text{top}}} \mathbf{p}[i] \leq \delta$. Let $S = \sum_{i \in k_{\text{top}}} \mathbf{p}[i]$. We denote $\bar{\mathbf{q}}$ as

the resulting probability vector normalized by the sum of the k_{top} values; more explicitly, $\bar{\mathbf{q}}[i] = \frac{\mathbf{p}[i]}{S}$ if $i \in k_{\text{top}}$ and zero otherwise. Lastly, $\mathbf{q}_{\text{SLQ}}(\mathbf{p})$ is the subsequent probability vector after passing the non-zero values of $\bar{\mathbf{q}}$ into the standard LQ algorithm (Algorithm 1).

To determine the number of bits needed to send the index of $\mathbf{q}_{\text{SLQ}}(\mathbf{p})$ we want to bound the bit budget as a function of the source distortion incurred. Under SLQ, there are two causes of distortion: normalization of the k_{top} highest values and standard LQ. To represent this, we first prove the following statement on the distortion encapsulated by both operations:

$$D_{TV}(\mathbf{p}, \mathbf{q}_{\text{SLQ}}(\mathbf{p})) \leq D_{TV}(\mathbf{p}, \bar{\mathbf{q}}) + D_{TV}(\bar{\mathbf{q}}, \mathbf{q}_{\text{SLQ}}(\mathbf{p})), \quad (34)$$

where $D_{TV}(\mathbf{p}, \bar{\mathbf{q}})$ represents the distortion incurred through normalization and $D_{TV}(\bar{\mathbf{q}}, \mathbf{q}_{\text{SLQ}}(\mathbf{p}))$ represents the distortion incurred through LQ. We prove (34) as follows:

$$\begin{aligned} D_{TV}(\mathbf{p}, \mathbf{q}_{\text{SLQ}}(\mathbf{p})) &= \frac{1}{2}|\mathbf{p} - \mathbf{q}_{\text{SLQ}}(\mathbf{p})| \\ &= \frac{1}{2}|\mathbf{p} - \bar{\mathbf{q}} + \bar{\mathbf{q}} - \mathbf{q}_{\text{SLQ}}(\mathbf{p})| \\ &\overset{(a)}{\leq} \frac{1}{2}|\mathbf{p} - \bar{\mathbf{q}}| + \frac{1}{2}|\bar{\mathbf{q}} - \mathbf{q}_{\text{SLQ}}(\mathbf{p})| \\ &= D_{TV}(\mathbf{p}, \bar{\mathbf{q}}) + D_{TV}(\bar{\mathbf{q}}, \mathbf{q}_{\text{SLQ}}(\mathbf{p})), \end{aligned}$$

where (a) follows from the triangle inequality. Having proved (34), we now upper-bound $D_{\text{TV}}(\mathbf{p}, \mathbf{q}_{\text{SLQ}}(\mathbf{p}))$ by the required source distortion β_s, which can be explicitly stated as:

$$D_{TV}(\mathbf{p}, \mathbf{q}_{\text{SLQ}}(\mathbf{p})) \leq D_{TV}(\mathbf{p}, \bar{\mathbf{q}}) + D_{TV}(\bar{\mathbf{q}}, \mathbf{q}_{\text{SLQ}}(\mathbf{p})). \quad (35)$$

We upper bound $D_{TV}(\mathbf{p}, \bar{\mathbf{q}})$ as follows:

$$\begin{aligned} D_{TV}(\mathbf{p}, \bar{\mathbf{q}}) &= \frac{1}{2}\sum_i |\mathbf{p}[i] - \bar{\mathbf{q}}[i]| \\ &= \frac{1}{2}\left(\sum_{i \in k_{\text{top}}} \left|\mathbf{p}[i] - \frac{\mathbf{p}[i]}{S}\right| + \sum_{i \notin k_{top}} |\mathbf{p}[i] - \bar{\mathbf{q}}[i]|\right) \\ &\overset{(a)}{=} \frac{1}{2}\left(\sum_{i \in k_{\text{top}}} \mathbf{p}[i]\left|1 - \frac{1}{S}\right| + \sum_{i \notin k_{\text{top}}} |\mathbf{p}[i]|\right) \\ &= \frac{1}{2}\left(\sum_{i \in k_{\text{top}}} \mathbf{p}[i]\left|\frac{S-1}{S}\right| + \sum_{i \notin k_{\text{top}}} |\mathbf{p}[i]|\right) \\ &\overset{(b)}{=} \frac{1}{2}\left(\sum_{i \in k_{\text{top}}} \mathbf{p}[i]\frac{|S-1|}{S} + \sum_{i \notin k_{\text{top}}} \mathbf{p}[i]\right) \end{aligned} \quad (36)$$

where (a) follows from $\bar{\mathbf{q}}[i] = 0 \ \forall i \notin k_{\text{top}}$, (b) follows from $S \geq 0$.

$$\begin{aligned} D_{TV}(\mathbf{p}, \bar{\mathbf{q}}) &\overset{(a)}{=} \frac{1}{2}\left(\sum_{i \in k_{\text{top}}} \mathbf{p}[i]\frac{1-S}{S} + \sum_{i \notin k_{\text{top}}} \mathbf{p}[i]\right) \\ &= \frac{1}{2}\left(\sum_{i \in k_{top}} \frac{\mathbf{p}[i]}{S} - \sum_{i \in k_{\text{top}}} \mathbf{p}[i] + \sum_{i \notin k_{top}} \mathbf{p}[i]\right) \\ &\overset{(b)}{=} \frac{1}{2}\left(1 - \sum_{i \in k_{\text{top}}} \mathbf{p}[i] + \sum_{i \notin k_{\text{top}}} \mathbf{p}[i]\right) \\ &\overset{(c)}{=} \left(1 - \sum_{i \in k_{\text{top}}} \mathbf{p}[i]\right) \\ &\overset{(d)}{\leq} \delta, \end{aligned} \quad (37)$$

where (a) follows from $0 \leq S \leq 1$, (b) follows from $S = \sum_{i \in k_{\text{top}}} \mathbf{p}[i]$, (c) follows from $\sum_{i \notin k_{\text{top}}} \mathbf{p}[i] = 1 - \sum_{i \in k_{\text{top}}} \mathbf{p}[i]$, (d) follows from $\sum_{i \notin k_{\text{top}}} \mathbf{p}[i] = 1 - \sum_{i \in k_{\text{top}}} \mathbf{p}[i] \leq \delta$.
From (8), we can upper bound the distortion due to LQ as $D_{TV}(\bar{\mathbf{q}}, \mathbf{q}_{\text{SLQ}}(\mathbf{p})) \leq \frac{k_{\text{top}}}{4\ell}$, as only the k_{top} non-zero values of $\bar{\mathbf{q}}$ will be passed into Algorithm 1 for quantization. Recalling (9), this would imply that $\left\lceil \log_2 \binom{\ell + k_{top} - 1}{k_{top} - 1} \right\rceil$ bits are needed to send the index of the resulting probability vector. Substituting these bounds into (35) gives

$$\delta + \frac{k_{\text{top}}}{4\ell} \leq \beta_s. \quad (38)$$

Solving for ℓ in (38) gives:

$$\ell = \left\lceil \frac{k_{\text{top}}}{4(\beta_s - \delta)} \right\rceil, \quad (39)$$

which implies that $\beta_s > \delta$. We now have a bound on the choice of ℓ to ensure a source distortion no greater than β_s accounting for normalization and standard LQ. The positions of the k_{top} highest predictions also need to be transmitted for the receiver to know which classes the probabilities correspond to. To address this, the set of positions of the k_{top} highest values are represented by generating a set of k bits where a 1 is used for an index if it is one of the k_{top} highest probabilities and a 0 is used otherwise. Subsequently, the k bits needed to represent the indices can be lower bounded by $\left\lceil \log_2 \binom{k}{k_{\text{top}}} \right\rceil$. Thus, $\left\lceil \log_2 \binom{\ell + k_{\text{top}} - 1}{k_{\text{top}} - 1} \right\rceil + \left\lceil \log_2 \binom{k}{k_{\text{top}}} \right\rceil$ bits under SLQ with $\ell = \left\lceil \frac{k_{\text{top}}}{4(\beta_s - \delta)} \right\rceil$ will satisfy the source distortion requirement.

Proof of Lemma 5

We can bound the end-to-end expected distortion as

$$\begin{aligned} \mathbb{E}[D_{\text{TV}}(\mathbf{p}, \hat{\mathbf{p}}(\kappa(\mathbf{y}))] &\overset{(a)}{=} P(\psi(\mathbf{p}) = \kappa(\mathbf{y}))D_1 + P(\psi(\mathbf{p}) \neq \kappa(\mathbf{y}))D_2 \\ &\overset{(b)}{\leq} (1 - \epsilon^*(n))D_1 + \epsilon^*(n)D_2 \\ &\overset{(c)}{\leq} (1 - \epsilon^*(n))\beta_s + \epsilon^*(n), \end{aligned} \quad (40)$$

where (a) follows from the total probability theorem, and $D_1 = \mathbb{E}[D_{\text{TV}}(\mathbf{p}, \psi(\mathbf{p}))|\psi(\mathbf{p}) = \kappa(\mathbf{y})]$ ($D_2 = \mathbb{E}[D_{\text{TV}}(\mathbf{p}, \hat{\mathbf{p}})|\psi(\mathbf{p}) \neq$

$\kappa(\mathbf{y})]$, respectively) is the expected distortion when quantized probability vector is exactly constructed (not exactly reconstructed, respectively) at the receiver; (b) follows from considering a bound on the decoding error probability such that, $\mathbb{P}(\psi(\mathbf{p}) \neq \kappa(\mathbf{y})) \leq \epsilon^*(n)$. (c) follows using the source distortion constraint $D_1 \leq \beta_s$, and from the fact that the total distortion is quantified using TV-divergence which allows us to bound $D_2 \leq 1$.

REFERENCES

[1] N. Teku, S. Adiga, R. Tandon, "Communicating Classification Results over Noisy Channels", To be published in 2024 IEEE International Conference on Communications (ICC).

[2] D.Choi, J. Yim, M. Baek, S. Lee, "Machine Learning-Based Vehicle Trajectory Prediction Using V2V Communications and On-Board Sensors.", Electronics 2021, vol. 10, no.4: 420.

[3] A. H. Sakr, G. Bansal, V. Vladimerou and M. Johnson, "Lane Change Detection Using V2V Safety Messages," 2018 21st International Conference on Intelligent Transportation Systems (ITSC), Maui, HI, USA, pp. 3967-3973, 2018.

[4] D. Gündüz et al., "Beyond Transmitting Bits: Context, Semantics, and Task-Oriented Communications," in IEEE Journal on Selected Areas in Communications, vol. 41, no. 1, pp. 5-41, Jan. 2023.

[5] S.Ma, et al. "A Theory for Semantic Communications." arXiv preprint arXiv:2303.05181 (2023).

[6] W. Yang et al., "Semantic Communications for Future Internet: Fundamentals, Applications, and Challenges," in IEEE Communications Surveys & Tutorials, vol. 25, no. 1, pp. 213-250, First quarter 2023.

[7] X. Luo, H. -H. Chen and Q. Guo, "Semantic Communications: Overview, Open Issues, and Future Research Directions," in IEEE Wireless Communications, vol. 29, no. 1, pp. 210-219, February 2022.

[8] G. Durisi, T. Koch and P. Popovski, "Toward Massive, Ultrareliable, and Low-Latency Wireless Communication With Short Packets," in Proceedings of the IEEE, vol. 104, no. 9, pp. 1711-1726, Sept. 2016.

[9] P. Popovski et al., "Wireless Access in Ultra-Reliable Low-Latency Communication (URLLC)," in IEEE Transactions on Communications, vol. 67, no. 8, pp. 5783-5801, Aug. 2019.

[10] M. Fresia, F. Perez-Cruz, H. V. Poor and S. Verdu, "Joint Source and ´ Channel Coding," in IEEE Signal Processing Magazine, vol. 27, no. 6, pp. 104-113, Nov. 2010.

[11] D. B. Kurka, and D. Gunduz. "Bandwidth-agile image transmission with deep joint source-channel coding.", in IEEE Transactions on Wireless Communications, vol. 20, no.12, pp. 8081-8095, Dec. 2021

[12] N. Farsad, R. Milind, and A. Goldsmith, "Deep learning for joint source-channel coding of text.", 2018 IEEE international conference on acoustics, speech and signal processing (ICASSP). IEEE, 2018.

[13] T.Tung, and D. Gunduz. "Deep joint source-channel and encryption coding: Secure semantic communications.", ICC 2023-IEEE International Conference on Communications. IEEE, 2023.

[14] H. Xie, Z. Qin, G. Y. Li and B. -H. Juang, "Deep Learning Enabled Semantic Communication Systems," in IEEE Transactions on Signal Processing, vol. 69, pp. 2663-2675, 2021.

[15] H. Xie, Z. Qin and G. Y. Li, "Semantic Communication With Memory," in IEEE Journal on Selected Areas in Communications, vol. 41, no. 8, pp. 2658-2669, Aug. 2023.

[16] Y. Xiao, X. Zhang, Y. Li, G. Shi and T. Başar, "Rate-Distortion Theory for Strategic Semantic Communication," 2022 IEEE Information Theory Workshop (ITW), Mumbai, India, pp. 279-284, 2022.

[17] J. Liu, W. Zhang and H. V. Poor, "A Rate-Distortion Framework for Characterizing Semantic Information," 2021 IEEE International Symposium on Information Theory (ISIT), Melbourne, Australia, pp. 2894-2899, 2021.

[18] S. Enayati and H. Pishro-Nik, "Quantization Rate and AoI-Induced Distortion Trade-off Analysis with Application to Remote Agents," 2022 IEEE Wireless Communications and Networking Conference (WCNC), Austin, TX, USA, pp. 782-787, 2022.

[19] Y. A. Reznik, "An Algorithm for Quantization of Discrete Probability Distributions," 2011 Data Compression Conference, Snowbird, UT, USA, pp. 333-342, 2011.

[20] Y. Reznik, V. Chandrasekhar, G. Takacs, D. M. Chen, S. S. Tsai and B. Girod, "Fast Quantization and Matching of Histogram-based Image Feature Descriptors," Proc. SPIE Vol. 7798, Applications of Digital Image Processing, pp. 779820-1–14, San Diego, August 2010.

[21] J. Conway and N. Sloane, "Fast quantizing and decoding and algorithms for lattice quantizers and codes," in IEEE Transactions on Information Theory, vol. 28, no. 2, pp. 227-232, March 1982.

[22] R.Cabasag, S. Huq, E. Mendoza, & M. K. Roychowdhury, "Optimal quantization for discrete distributions", 2020, arXiv preprint arXiv:2008.03255.

[23] S. Matsuura and K. Hiroshi, "Statistical Estimation of Quantization for Probability Distributions: Best Equivariant Estimator of Principal Points." International Conference on Machine Learning, Optimization, and Data Science. Cham: Springer International Publishing, 2021.

[24] A. Adler, J. Tang and Y. Polyanskiy, "Quantization of Random Distributions under KL Divergence," 2021 IEEE International Symposium on Information Theory (ISIT), Melbourne, Australia, pp. 2762-2767, 2021.

[25] A. Adler, J. Tang and Y. Polyanskiy, "Efficient Representation of Large-Alphabet Probability Distributions," in IEEE Journal on Selected Areas in Information Theory, vol. 3, no. 4, pp. 651-663, Dec. 2022.

[26] W. Liu, G. Nair, Y. Li, D. Nesic, B. Vucetic and H. V. Poor, "On the Latency, Rate, and Reliability Tradeoff in Wireless Networked Control Systems for IIoT," in IEEE Internet of Things Journal, vol. 8, no. 2, pp. 723-733, 15 Jan.15, 2021.

[27] D. Qiao, M. C. Gursoy and S. Velipasalar, "Throughput-Delay Tradeoffs With Finite Blocklength Coding Over Multiple Coherence Blocks," in IEEE Transactions on Communications, vol. 67, no. 8, pp. 5892-5904, Aug. 2019.

[28] S. Wang, Y. Yuan, S. Lv, J. Jin, Q. Wang and G. Liu, "Optimal Tradeoff between Reliability and Latency with Finite Blocklength for Many Access Channel," 2021 IEEE Wireless Communications and Networking Conference Workshops (WCNCW), Nanjing, China, pp. 1-6, 2021.

[29] W. Cheng, Y. Xiao, S. Zhang and J. Wang, "Adaptive Finite Blocklength for Ultra-Low Latency in Wireless Communications," in IEEE Transactions on Wireless Communications, vol. 21, no. 6, pp. 4450-4463, June 2022.

[30] A. Rényi, "On measures of entropy and information." Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Contributions to the Theory of Statistics. Vol. 4. University of California Press, 1961.

[31] A. Lancho, T. Koch and G. Durisi, "On Single-Antenna Rayleigh Block-Fading Channels at Finite Blocklength," in IEEE Transactions on Information Theory, vol. 66, no. 1, pp. 496-519, Jan. 2020.

[32] https://github.com/geifmany/cifar-vgg

[33] https://colab.research.google.com/github/kundajelab/label_shift_experiments/blob/master/cifar100/CIFAR100_Compute_Predictions.ipynb

[34] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556, 2014

[35] S. Liu and W. Deng, "Very deep convolutional neural network based image classifi- cation using small training sample size.", 3rd IAPR Asian conference on pattern recognition (ACPR), pages 730–734, IEEE, 2015.

[36] https://learnopencv.com/image-classification-pretrained-imagenet-models-tensorflow-keras/

[37] K.He, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[38] J. Carreira & A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset." In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 6299-6308, 2017.

[39] https://www.tensorflow.org/hub/tutorials/action_recognition_with_tf_hub

[40] O. Russakovsky*, J. Deng*, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg and L. Fei-Fei. (* = equal contribution) "ImageNet Large Scale Visual Recognition Challenge" arXiv:1409.0575, 2014.

[41] A. Krizhevsky, "Learning multiple layers of features from tiny images." 2009.

[42] W. Kay, et. al. "The kinetics human action video dataset" arXiv preprint arXiv:1705.06950, 2017.

[43] K. Soomro, A. Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild." arXiv preprint arXiv:1212.0402, 2012.

[44] Y. Polyanskiy, H. V. Poor and S. Verdu, "Channel Coding Rate in the Finite Blocklength Regime," in IEEE Transactions on Information Theory, vol. 56, no. 5, pp. 2307-2359, May 2010.

[45] https://pytorch.org/tutorials/beginner/blitz/cifar10_tutorial.html

[46] W. F. Lo, N. Mital, H. Wu and D. Gündüz, "Collaborative Semantic Communication for Edge Inference," in IEEE Wireless Communications Letters, vol. 12, no. 7, pp. 1125-1129, July 2023.

Noel Teku earned his B.S. and M.S. in Electrical Engineering from the University of Arizona in 2016 and 2017, where he is also pursuing a Ph.D. in Electrical Engineering. His research interests include wireless communications, signal processing, and machine learning. He also served as an intern at Qualcomm in 2023.

Sudarshan Adiga earned his B.E. degree in Telecommunication Engineering from the Ramaiah Institute of Technology, Bangalore, in 2015, and his M.S. and Ph.D. degrees in Electrical and Computer Engineering from the University of Arizona, Tucson, USA, in 2019 and 2024, respectively. His research interests span machine learning, information theory, and wireless communications. He currently works as a Staff Engineer at Marvell Technology. Prior to his doctoral studies, Sudarshan worked as a software engineer at Bosch Corporation in India and Japan from 2015 to 2017. He also served as a research intern at NTT Docomo in 2022 and Marvell Technology in 2023.

Ravi Tandon (Senior Member, IEEE) received the B.Tech. degree in electrical engineering from the Indian Institute of Technology, Kanpur (IIT Kanpur), in 2004, and the Ph.D. degree in electrical and computer engineering (ECE) from the University of Maryland, College Park (UMCP), in 2010. He is currently the Litton Industries John M. Leonis Distinguished Associate Professor with the Department of ECE, The University of Arizona. Prior to joining The University of Arizona in Fall 2015, he was a Research Assistant Professor with Virginia Tech. He held a positions with the Bradley Department of ECE, Hume Center for National Security and Technology; and the Department of Computer Science, Discovery Analytics Center. From 2010 to 2012, he was a Post-Doctoral Research Associate with Princeton University. His current research interests include information theory and its applications to machine learning, privacy, security, wireless networks and signal processing. He was a recipient of the 2018 Keysight Early Career Professor Award, the NSF CAREER Award in 2017, and a Best Paper Award at IEEE GLOBECOM 2011. He has served on the editorial board of IEEE Transactions on Wireless Communications and is currently an editor for IEEE Transactions on Communications and IEEE Transactions on Information Theory.