Task-Oriented Communication for Edge Video Analytics

Jiawei Shao®, Graduate Student Member, IEEE, Xinjie Zhang, Graduate Student Member, IEEE, and Jun Zhang®, Fellow, IEEE

Abstract—With the development of artificial intelligence (AI) techniques and the increasing popularity of camera-equipped devices, many edge video analytics applications are emerging, calling for the deployment of computation-intensive AI models at the network edge. Edge inference is a promising solution to move computation-intensive workloads from low-end devices to a powerful edge server for video analytics, but device-server communications will remain a bottleneck due to limited bandwidth. This paper proposes a task-oriented communication framework for edge video analytics, where multiple devices collect the visual sensory data and transmit the informative features to an edge server for processing. To enable low-latency inference, this framework removes video redundancy in spatial and temporal domains and transmits minimal information that is essential for the downstream task, rather than reconstructing the videos on the edge server. Specifically, it extracts compact task-relevant features based on the deterministic information bottleneck (IB) principle, which characterizes a tradeoff between the informativeness of the features and the communication cost. As the features of consecutive frames are temporally correlated, we propose a temporal entropy model (TEM) to reduce the bitrate by taking the previous features as side information in feature encoding. To further improve the inference performance, we build a spatial-temporal fusion module on the server to integrate features of the current and previous frames for joint inference. Extensive experiments on video analytics tasks evidence that the proposed framework effectively encodes task-relevant information of video data and achieves a better rate-performance tradeoff than existing methods.

Index Terms—Task-oriented communication, edge video analytics, deterministic information bottleneck, temporal entropy model.

I. INTRODUCTION

VIDEO analytics has grown substantially from practical needs in the past decade, driven by the revival of artificial intelligence (AI) and the advancement of deep learning. AI-enabled video analytics applications are being widely deployed across different fields such as smart transportation [1], security surveillance systems [2], and online healthcare [3]. The main goal is to automatically recognize temporal-spatial events in videos and gain valuable insights. At the core of these applications lie powerful deep learning models performing tasks including image classification, object detection, and semantic segmentation.

As edge devices equipped with camera sensors (e.g., drones, mobile phones, and monitoring cameras) are becoming increasingly widespread, the industry has shown strong interest in deploying video analytics at the network edge to support ubiquitous AI-enabled services, which gives rise to a new research area named edge video analytics [4]. As the deep learning models demand intensive computation, leading to a long delay when running on resource-constrained edge devices, typical edge systems forward all the input data from the edge devices to a proximate edge server (e.g., a base station) to perform inference. However, when devices are connected wirelessly, such as wearable devices and autonomous cars, the available uplink bandwidth is very limited and the communication channel is highly dynamic. The enormous volume of the collected videos leads to prohibitive communication overhead if directly transmitting them to the edge server. Meanwhile, many video analytics applications have stringent response time requirements, the high communication latency hinders user experience. This tension between the communication overhead and the latency requirement brings the main design challenge of edge video analytics systems.

To reduce the bandwidth consumption and response time in edge video analytics, a cooperative edge inference framework stands out as a promising solution [5], [6]. With multiple devices at the network edge that capture video data, each device first extracts compact features from input frames that preserve the task-relevant information and then forwards them to be processed by an edge server. This exemplifies a recent shift in communication system design to support intelligence applications at the edge, named task-oriented communication [6], [7], [8], [9], [10], [11], [12], which optimizes communication strategies for the downstream task, rather than for recovering data at the server as in traditional data-oriented communication. Many existing works have led to efficient communication strategies for edge inference. For example, the VFE method proposed in [7] formalized a tradeoff between the communication rate and accuracy based on the information bottleneck (IB) principle [13] for single-device edge inference. Later, the authors of [6] considered the task-oriented communication framework for multi-view computer vision tasks in edge inference. They developed a distributed feature encoding
method to reduce the communication volumes, which squeezes the spatial redundancy among the extracted features of multi-view images.

While cooperative edge inference with task-oriented communication has enabled low-latency edge applications in previous works, directly applying these methods to edge video analytics may lose their effectiveness due to two main reasons. First, identifying the task-relevant features from videos requires a high-complexity extractor, which needs to process a sequence of frames by either recurrent neural networks [14] or attention-based transformer blocks [15]. These two AI models demand much higher computation and memory resources than convolutional neural networks [16] adopted in previous studies on image-related tasks [6], [7]. Second, previous works neglected the copious temporal information during the feature encoding process, which resulted in high redundancy for video data. In many applications such as pedestrian tracking [17] and vehicle re-identification [18], it is intuitive that motion estimation [19] can help to exploit the temporal correlation among adjacent frames to substantially reduce the communication cost since the backgrounds of consecutive frames are visually similar. This observation motivates us to develop a task-oriented communication framework to tackle the aforementioned problems and accelerate video analytics at the network edge. Particularly, we leverage the deterministic information bottleneck principle [20] to extract the task-relevant feature from each frame independently and develop a spatial-temporal fusion module at the server to jointly leverage the current received features as well as the previous features to perform inference. The temporal entropy model helps to reduce the encoded feature size by using the previously transmitted features as side information.

A. Related Works

The recent investigation of task-oriented communication [6], [21], [22], [23], [24], [25], [26], as well as similar ideas such as semantic communication [27], [28], has motivated a paradigm shift in communication system design. The conventional data-oriented communication aims to guarantee the reliable transmission of every single bit of the raw data, while oblivious to the message semantics or the downstream task. For edge video analytics, it will incur excessive delay if the high-volume video data is directly transmitted with the limited communication resources at the network edge. The task-oriented communication principle is promising to alleviate this challenge. Specifically, it discards information of the input frames (e.g., image noise and background) that is irrelevant to the given task, and transmits only the task-relevant feature (e.g., region of interest) that could influence the inference result. In this way, it has the potential to achieve orders of magnitude reduction in the communication overhead, thus significantly reducing the transmission latency.

There have been several recent studies taking advantage of deep learning to develop effective feature extraction and encoding methods following the task-oriented design principle [6], [7], [29], [30], [31], [32]. In particular, an end-to-end architecture for image classification was proposed in [33], which leverages a JPEG compressor to encode the internal features for the downstream task. The proposed method was directly trained with the loss function specified by the task and did not take the feature recovery quality into consideration. The same idea was utilized in other intelligence edge applications including image retrieval [34], point cloud processing [35], and semantic communication [27]. Besides, the authors of [7] formalized a tradeoff between the communication rate and the inference performance in task-oriented communication based on the information bottleneck (IB) principle [13]. Notably, the IB principle provides the insight that a good feature should be sufficient for the inference task while retaining minimal information from the input data. This is consistent with the design objective in cooperative edge inference. Meanwhile, the IB principle is also adopted in [21] to identify the relevant semantic information to accomplish a specific goal. The most recent work [6] extended this task-oriented communication framework to multi-device edge inference systems, which developed a distributed feature coding method to squeeze the spatial redundancy among the extracted features of multiple devices.

Although the aforementioned works proposed several task-oriented communication strategies to reduce the communication overhead at the network edge, they mainly considered simple tasks with a single instance as the input data (e.g., image classification). Directly applying them to tasks with sequential data (e.g., edge video analytics systems) is not efficient. This is because the methods designed for single-frame processing cannot exploit the temporal correlation among a sequence of frames. Traditional video compression algorithms [36], [37] adopt a predictive coding architecture to achieve a high compression ratio. Particularly, these methods apply a motion-compensation loop at the encoder to explicitly investigate the inter-frame redundancy. Recently, deep learning-based video compression methods have been developed [38], [39], [40], [41], [42], [43], [44], [45], which enable end-to-end optimization to further improve the compression performance. However, both the traditional and learning-based video codecs require a heavy encoder to exploit the statistical redundancy in the pixel domain, aiming at recovering the video with high resolution under a bitrate constraint. This kind of data-oriented objective does not fit well with video analytics applications at the edge, which motivates us to develop a task-oriented communication framework to exploit the temporal-spatial correlation of videos in the feature domain. Meanwhile, we will also take inspiration from deep learning-based compression methods.

B. Contributions

In this paper, we develop a task-oriented communication framework for edge video analytics, which effectively extracts task-relevant features and reduces both the spatial and temporal redundancy in the feature domain. The major contributions are summarized as follows:

- This work formulates three design problems of task-oriented communication for edge video analytics, namely, task-relevant feature extraction, feature encoding,
and joint inference. We propose a task-oriented communication (TOCOM) framework with a temporal entropy model (TEM) for edge video analytics, which is named TOCOM-TEM.

- First, we leverage the deterministic information bottleneck to design the local feature extractor, which aims at maximizing the mutual information between the extracted feature and the desired output while minimizing the entropy of the feature. Thus, it addresses the objective of reducing communication overhead by discarding task-irrelevant information. As the mutual information and entropy terms in the deterministic information bottleneck are computationally prohibitive for high-dimensional data, we resort to variational approximations to devise tractable upper bounds.
- Second, to further compress the size of the extracted feature and reduce the joint bitrate, we perform entropy coding to encode the current feature by adopting the previous feature as side information. We proposed a temporal entropy model to exploit the dependence among features by estimating the conditional probability distribution of the current feature given the previous ones.
- Third, to better exploit both spatial and temporal correlation of video data collected from multiple devices, we construct a spatial-temporal fusion module at the server to integrate the current received features and the previous features to predict the desired output.
- The effectiveness of the proposed framework is validated in two edge video analytics tasks, i.e., a multi-camera pedestrian occupancy prediction task and a multi-camera object detection task. Extensive simulation results show that our TOCOM-TEM method achieves a substantial reduction in the communication overhead compared with the image compression and video coding methods. Our study demonstrates that the task-oriented design principle is promising to enable low-latency edge video analytics applications.

C. Organization

The rest of this paper is organized as follows. Section II introduces the system model and formulates the design problems of task-oriented communication for edge video analytics, and Section III proposes the TOCOM-TEM method. In Section IV, we provide extensive simulation results to evaluate the performance of the proposed methods. Finally, Section V concludes the paper.

D. General Notations

Throughout this paper, upper-case letters (e.g., X, Y) and lower-case letters (e.g., x, y) stand for random variables and their realizations, respectively. We abbreviate a sequence \( \{X_1, \ldots, X_N\} \) of \( N \) random variables by \( X_{1:N} \) and similarly use \( X^{(1:K)} \) to represent the set of random variables \( \{X^{(1)}, \ldots, X^{(K)}\} \). The entropy of \( Y \), the conditional entropy of \( Y \) given \( X \), and the joint entropy of \( X, Y \) are denoted as \( H(Y) \), \( H(Y|X) \), and \( H(X, Y) \), respectively. The mutual information between \( X \) and \( Y \) is represented as \( I(X, Y) \). The Kullback-Leibler (KL) divergence between two probability distributions \( p(x) \) and \( q(x) \) is denoted as \( D_{KL}(p||q) \), and the cross entropy between \( p(x) \) and \( q(x) \) is represented as \( H(p, q) \). The statistical expectation of \( X \) is denoted as \( E(X) \).

II. SYSTEM MODEL AND PROBLEM FORMULATION

A. System Model

We consider an edge video analytics system, where multiple devices collect video sensing data from the environment and forward them to an edge server to output the inference result. Such edge video analytics can effectively overcome the limitations of resource-constrained devices, as the server has abundant computational resources to efficiently execute the inference task. Besides, leveraging the spatial-temporal cues from multi-view videos overcomes the limited sensing capability of a single device, captures the dynamic patterns among consecutive frames, and improves inference performance. Related application scenarios include multi-camera tracking [46] and vehicle re-identification [18]. A sample system for pedestrian occupancy prediction [47] is shown in Fig. 1.

The main challenge of edge inference systems is the limited communication bandwidth, which makes it infeasible to transmit high-volume raw videos to the edge server. Besides, minimizing the communication overhead is also desirable to save device energy. Thus, we will adopt the task-oriented communication principle [48] that only transmits the information that is essential to the downstream tasks to the edge server for inference. Compared with image-based tasks investigated in [6] and [7], video analytics tasks are more challenging since the input data are temporally correlated. The following subsections first present a probabilistic modeling of the edge video analytics system and then formulate the design problems for task-oriented communication.

B. Probabilistic Modeling

We consider an edge video analytics system comprising \( K \) devices as shown in Fig. 2. The consecutive frames of each device \( k \in 1:K \) with length \( N \) are denoted as \( x^{(k)}_{1:N} := \{x^{(k)}_1, \ldots, x^{(k)}_N\} \). At each time step \( t \in 1:N \), the input views captured by different devices are denoted by \( x^{(1:K)}_t := \{x^{(1)}_t, \ldots, x^{(K)}_t\} \), which is deemed as realizations of random variables \( X^{(1:K)}_t := \{X^{(1)}_t, \ldots, X^{(K)}_t\} \).

The edge server aims to predict the desired output \( y_{1:N} := \{y_1, \ldots, y_N\} \) based on the frames collected by the devices. In particular, we define the random variables of the output as \( Y_{1:N} := \{Y_1, \ldots, Y_N\} \). As not every part of the input data \( x^{(k)}_t \) is relevant to the inference task, the devices should identify the useful information \( z^{(k)}_t \) and discard the redundancy to reduce the communication overhead. At each time step \( t \), the device \( k \) extracts the task-relevant feature \( z^{(k)}_t \) from its
Fig. 1. An example of edge video analytics for pedestrian occupancy prediction. The edge server takes consecutive video frames from multiple cameras as inputs to estimate the position of pedestrians.

Fig. 2. Probabilistic modeling for edge video analytics systems. At each time step $t$, edge devices extract the task-relevant features $z_{(k)}^t$ from their inputs $x_{(k)}^t$, adopt the temporal entropy model to encode the quantized feature $\hat{z}_{(k)}^t$, and forward them to an edge server for further processing. A spatial-temporal fusion module is deployed at the edge server that jointly leverages the current received features $\hat{z}_{(1:K)}^t$ as well as the previous features $\{\hat{z}_{(1:K)}^{t-1}, \ldots , \hat{z}_{(1:K)}^{t-\tau_1}\}$ to predict the target variable $\hat{y}_t$.

input data $x_{(k)}^t$ and then rounds the feature to $\hat{z}_{(k)}^t = \lfloor z_{(k)}^t \rfloor$. Particularly, the extracted features $z_{1:N}^{(k)} := \{z_{1}^{(k)}, \ldots , z_{N}^{(k)}\}$ and the discrete features $\hat{z}_{1:N}^{(k)} := \{\hat{z}_{1}^{(k)}, \ldots , \hat{z}_{N}^{(k)}\}$ are regarded as the instantiations of random variables $Z_{1:N}^{(k)} := \{Z_{1}^{(k)}, \ldots , Z_{N}^{(k)}\}$ and $\hat{Z}_{1:N}^{(k)} := \{\hat{Z}_{1}^{(k)}, \ldots , \hat{Z}_{N}^{(k)}\}$, respectively. At the network edge, the server receives the features from the devices to predict the target variables. The key design objective is to minimize the communication overhead in feature transmission while providing satisfactory inference performance.

C. Problem Formulation

Learning a task-oriented communication framework for edge video analytics can be decomposed into three subproblems, including feature extraction, feature encoding, and joint inference.

1) Feature Extraction: At each time slot $t$, the extracted feature $\hat{z}_{(k)}^t$ at device $k$ is transmitted to the edge server. By extracting more task-relevant information from the input frame, the edge server holds the potential to improve the inference performance. However, this comes at the cost of increased communication overhead. So the extracted features should be maximally informative about the target variable $y_t$ (i.e., to promote inference performance), subject to a constraint on its representation complexity (i.e., to control the communication overhead). In this work, we resort to the deterministic information bottleneck to characterize the inherent tradeoff between the rate and inference performance. Particularly, the mutual information $I(Y_t; \hat{Z}_{1:N}^{(k)})$ is adopted to measure the informativeness between the target $Y_t$ and the
feature \( \hat{Z}_t^{(k)} \). The entropy \( H(\hat{Z}_t^{(k)}) \) is utilized to represent the complexity of \( \hat{Z}_t^{(k)} \). In edge video analytics, the ideal feature \( \hat{Z}_t^{(k)} \) should satisfy \( I(Y_t; \hat{Z}_t^{(k)}) = I(Y_t; X_{\leq t}^{(k)}), \) where \( X_{\leq t}^{(k)} := \{ X_1^{(k)}, \ldots, X_t^{(k)} \} \). This implies that the feature extractor needs to identify the task-relevant information from both the current frame as well as the previous frames.

However, given the limited resources at the devices, taking all the frames \( x_{t}^{(k)} \) as inputs for feature extraction is computationally prohibitive and will also induce excessive memory consumption. To alleviate this problem, we simplify the feature extraction process. Each device \( k \in 1:K \) extracts the feature \( z_t^{(k)} \) only from the current frame \( x_t^{(k)} \) using the function \( z_t^{(k)} = f(x_t^{(k)}; \theta^{(k)}) \) with parameters \( \theta^{(k)} \). The discrete feature \( \hat{z}_t^{(k)} \) is expected to discard the task-irrelevant redundancy and preserve the information about the current target variable \( y_t \) as well as the temporal clues that help to predict the next \( \tau_1 \) variables \( \{ y_{t+1}, \ldots, y_{t+\tau_1} \} \). This suggests the objective:

\[
\min_{\theta^{(1:K)}} \sum_{t=1}^{N} I(Y_t, \ldots, Y_{t+\tau_1}; \hat{Z}_t^{(1:K)}) + \beta \sum_{t=1}^{N} \sum_{k=1}^{K} H(\hat{Z}_t^{(k)}),
\]

(1)

where the parameter \( \beta > 0 \) trades off between the inference performance and the entropies of features, and \( \theta^{(1:K)} \) denotes the parameters \{\( \theta^{(1)}, \ldots, \theta^{(K)} \)\}. The first term represents the inference performance, and the second term denotes the amount of information preserved by the features. As minimizing the objective function in (1) requires high computation cost to estimate the distributions \( p(y_t | \hat{Z}_t^{(1:K)}) \) and \( p(\hat{Z}_t^{(k)}) \), we adopt variational approximation [49] to reformulate it into an amenable form in Section III-A.

2) Feature Encoding: After the feature extraction process, the second step is to encode the resulting discrete features for efficient transmission. To losslessly reconstruct the features at the server, we adopt the entropy coding that relies on a probability model (i.e., an entropy model) of the discrete variable to estimate the marginal distribution of the features. Denote the probability distribution of \( \hat{z}_t^{(k)} \) as \( p(\hat{z}_t^{(k)}) \) and the estimated distribution of the entropy model as \( q(\hat{z}_t^{(k)}) \). Entropy coding is able to encode \( \hat{z}_t^{(k)} \) at the bitrate of the cross-entropy \( H(p(\hat{z}_t^{(k)}), q(\hat{z}_t^{(k)})) \) with negligible overhead. The better the entropy model predicts the distribution, the fewer bits are required to represent the discrete feature. To effectively reduce the communication overhead in feature transmission, an intuitive method is to encode every subsequent feature \( \hat{z}_t^{(k)} \) depending on the previous features \( \hat{z}_{t-1}^{(k)} \) to exploit the temporal dependence. Therefore, we develop a temporal entropy model to estimate the conditional distribution of the current features given the previous ones. To reduce the modeling complexity, we make a \( \tau_2 \)-th order Markov assumption such that each feature \( \hat{z}_t^{(k)} \) for \( k \in 1:K \) only depends on the previous \( \tau_2 \) features \( \{ \hat{z}_{t-1}^{(k)}, \ldots, \hat{z}_{t-\tau_2}^{(k)} \} \), i.e., \( p(\hat{z}_t^{(k)} | \hat{z}_{t-1}^{(k)}, \ldots, \hat{z}_{t-\tau_2}^{(k)}) \). Define \( p(\hat{z}_t^{(k)} | \hat{z}_{t-1}^{(k)}, \ldots, \hat{z}_{t-\tau_2}^{(k)}) \) and \( q(\hat{z}_t^{(k)} | \hat{z}_{t-1}^{(k)}, \ldots, \hat{z}_{t-\tau_2}^{(k)}; \psi_p^{(k)}) \) as the true and estimated conditional distributions of \( \hat{z}_t^{(k)} \). In particular, the temporal entropy model of device \( k \) is parameterized by \( \psi_p^{(k)} \).

The conditional cross-entropy

\[
H(p(\hat{z}_t^{(k)} | \hat{z}_{t-1}^{(k)}, \ldots, \hat{z}_{t-\tau_2}^{(k)}); q(\hat{z}_t^{(k)} | \hat{z}_{t-1}^{(k)}, \ldots, \hat{z}_{t-\tau_2}^{(k)}; \psi_p^{(k)}))
\]

is expected to achieve a lower bitrate compared with the independent coding schemes. In our framework, each device \( k \) minimizes the conditional cross-entropy over the parameters \( \psi_p^{(k)} \) to reduce the communication overhead. More details about the training process have been deferred to Section III-B.

3) Joint Inference: Once the edge server has received the extracted features from devices, it can output the inference result based on the conditional distribution of the target variable given these features. As they are temporally correlated, it is expected that jointly leveraging both the current received features \( \hat{z}_t^{(1:K)} \) as well as \( \tau_1 \) previous features \( \{ \hat{z}_{t-\tau_1}^{(1:K)} \} \) reduces

| Notation | Description |
| --- | --- |
| \( x_t^{(k)} \) | Video frame of device \( k \) at time step \( t \). |
| \( f(x_t^{(k)}, \theta^{(k)}) \) | Feature extractor at device \( k \) with parameters \( \theta^{(k)} \). |
| \( z_t^{(k)} \) | Extracted feature and quantized feature. |
| \( h_u(z_t^{(k)}, \psi_u^{(k)}) \) | Hyper encoder at device \( k \) with parameters \( \psi_u^{(k)} \). |
| \( \psi_y^{(k)} \) | Hyper latent variable and quantized variable. |
| \( g(v_t^{(k)}, \psi_y^{(k)}) \) | Factorized entropy model parameterized by \( \psi_y^{(k)} \). |
| \( h_d(v_t^{(k)}, \psi_d^{(k)}) \) | Hyper decoder at device \( k \) with parameters \( \psi_d^{(k)} \). |
| \( g(z_t^{(k)}, \psi_d^{(k)}) \) | Variational approximation to \( p(z_t^{(k)} | v_t^{(k)}) \). |
| \( R_m \) | Bit constraint in loss \( Z_1 \). |
| \( \tau_1 \) | Temporal parameter of the feature extractor and the temporal-spatial fusion module. |
| \( \tau_2 \) | Temporal parameter of the entropy model. |
| \( g(z_t^{(1:K)}, \psi_r) \) | The \( \tau \)-th auxiliary predictor with parameters \( \varphi_r \). |
| \( g(y_{t+1}, \ldots, y_{t+\tau_1} | z_t^{(1:K)}, \psi_{y0:1}) \) | Variational approximation to \( p(y_{t+1}, \ldots, y_{t+\tau_1} | z_t^{(1:K)}) \). |
| \( h_p(z_t^{(1:K)}, \psi_p^{(k)}) \) | Parametric transform with parameters \( \psi_p^{(k)} \) in the temporal entropy model. |
| \( g(\hat{z}_t^{(k)} | \hat{z}_{t-1}^{(k)}, \ldots, \hat{z}_{t-\tau_2}^{(k)}; \psi_p^{(k)}) \) | Variational approximation to \( p(\hat{z}_t^{(k)} | \hat{z}_{t-1}^{(k)}, \ldots, \hat{z}_{t-\tau_2}^{(k)}) \). |
| \( g(z_t^{(1:K)}, \ldots, z_t^{(1:K)}; \psi_p^{(k)}) \) | Spatial-temporal fusion module with parameters \( \phi \). |
the uncertainty in inference. However, the distribution \( p(y_1 | z_{t-1}^{(1:K)}, \ldots, z_t^{(1:K)}) \) is difficult to compute due to the high-dimensional integral. We apply approximate inference methods by setting up a spatial-temporal fusion module \( y_t = g(z_t^{(1:K)}, \ldots, z_{t-1}^{(1:K)}; \phi) \) with parameters \( \phi \), which takes the features \( \{z_1^{(1:K)}, \ldots, z_{t-1}^{(1:K)}\} \) as input to output the inference result \( \hat{y}_t \). Let \( d(y_t, \hat{y}_t; \phi) \) represent a metric to measure the distortion between the predicted output \( \hat{y}_t \) and the target \( y_t \). The parameters \( \phi \) are optimized by minimizing this distortion function. The detailed training process is illustrated in Section III-C.

III. TASK-ORIENTED COMMUNICATION WITH TEMPORAL ENTROPY CODING

We propose a task-oriented communication framework with a temporal entropy model (TOCOM-TEM) for edge video analytics. It contains three components including task-relevant feature extraction, temporal entropy model, and spatial-temporal fusion module. First, each device \( k \in 1 : K \) extracts the task-relevant feature \( \hat{z}_t(k) \) from the current input \( x_t(k) \), and then applies entropy coding to compress the quantized feature \( \hat{z}_t(k) \) to a bitstream before transmitting it to the edge server. As the entropy coding relies on a probability model of the discrete variable, our method constructs a temporal entropy model that approximates the conditional distribution \( p(\hat{z}_t(k) | \hat{z}_{t-1}(1:K), \ldots, \hat{z}_{t-\tau}(1:K)) \) to reduce the bitrate by exploiting the relationships among features \( \{\hat{z}_1(k), \ldots, \hat{z}_{t-\tau}(k)\} \). Finally, a spatial-temporal fusion module at the edge server jointly leverages the current received features \( \hat{z}_t(k) \) as well as the previous features \( \{\hat{z}_{t-1}(1:K), \ldots, \hat{z}_{t-\tau}(1:K)\} \) to predict the target variable \( \hat{y}_t \). This section elaborates on the details of these three components.

A. Task-Relevant Feature Extraction

The first step is to extract the task-relevant features from the video frames. The loss function in (1) aims at maximizing the informativeness of the extracted features, while minimizing its communication overhead. However, directly optimizing the feature extractors based on this loss faces two challenges. One is that the gradients with respect to the parameters \( \theta^{(1:K)} \) are zero almost everywhere due to the quantization. The other is that taking the derivatives of the mutual information and the entropy is generally intractable since estimating the distributions \( p(y_1, \ldots, y_{t+1} | \hat{z}_t^{(1:K)}) \) and \( p(\hat{z}_t^{(1:K)}) \) are computationally prohibitive. In the following, we adopt continuous relaxation and variational approximation to alleviate these two problems, respectively.

1) Continuous Relaxation: To allow optimization via gradient descent type algorithms in the presence of quantization, we follow the noise-based relaxation in [50] to enable backpropagation. The main idea is replacing the quantization operation with additive uniform noise \( U(-0.5, 0.5) \). Denote the \( i \)-th element of the extracted feature \( z_t(k) \) as \( \hat{z}_{t,i}(k) \). The probability density function of \( \hat{z}_{t,i}(k) = \hat{z}_{t,i}(k) + U(-0.5, 0.5) \) is a continuous relaxation of the probability mass function of \( \hat{z}_{t,i}(k) = \hat{z}_{t,i}(k) \) such that \( p(\hat{z}_{t,i}(k) = \hat{z}) = p(\hat{z}_{t,i}(k) = \hat{z}) \) for integer \( \hat{z} \in \mathbb{Z} \). This implies that the probability density function of \( \hat{z}_{t,i}(k) \) can be used to accurately calculate the entropy of \( \hat{z}_{t,i}(k) \), and the differential entropy of variable \( z \) is a continuous proxy for the entropy of \( \hat{z}_{t,i}(k) \) in the training process.

2) Variational Approximation: Given the parameters \( \theta^{(1:K)} \), the distributions \( p(y_1, \ldots, y_{t+1} | \hat{z}_t^{(1:K)}) \) and \( p(\hat{z}_t^{(1:K)}) \) are fully determined by the data distribution \( p(x_1:N, y_1:N) \) and the feature extraction function \( f(x_t(k), \theta^{(k)}) \). In cases where the data distribution is neither known perfectly nor can be estimated to high accuracy, minimizing the objective function in (1) is nearly intractable. In this situation, we resort to the variational approximation method to formulate a tractable lower bound. The key idea is to posit a family of distributions and find a member of that family which is close to the target distribution.

To approximate the distribution \( p(y_1, \ldots, y_{t+1} | \hat{z}_t^{(1:K)}) \), we construct \( \tau_1 + 1 \) auxiliary predictors \( g(z_1^{(1:K)}; \varphi_1) \) for \( \tau \in \{0, \ldots, \tau_1\} \). As shown in Fig. 3, these predictors infer the future output \( y_{t+\tau} \) and \( y_{t+\tau+1} \). We formulate the variational distribution as \( q(y_1, \ldots, y_{t+\tau+1} | \hat{z}_t^{(1:K)}; \varphi_0, \ldots, \varphi_{\tau_1}) \propto \exp(-\sum_{\tau=0}^{\tau_1} w_{\tau}d(y_{t+\tau}, y_{t+\tau+1}; \varphi_{\tau})) \), where the scalars \( \{w_0, \ldots, w_{\tau_1}\} \) are weighting coefficients. Note that the joint entropy \( H(Y_1, \ldots, Y_{t+\tau}; \hat{Z}_t^{(1:K)}) \) is a constant. Minimizing the cross entropy between the true and variational distributions is equivalent to minimizing the upper bound of \(-I(Y_1, \ldots, Y_{t+\tau}; \hat{Z}_t^{(1:K)})\) in (1).

Besides, we construct \( K \) entropy models as variational approximations of the distributions \( p(\hat{z}_t^{(k)}) \) for \( k \in 1 : K \). These models are built over the hierarchical entropy model [50], which introduces a hyperprior to better capture the correlation among the elements in the extracted feature. As shown in Fig. 3, hyper encoder \( h_e^{(k)}(\hat{z}_t^{(k)}; \psi^{(k)}) \) transforms the extracted feature \( \hat{z}_t^{(k)} \) to a hyper-latent \( \hat{v}_t^{(k)} \), where \( \psi^{(k)} \) represents the neural network parameters. Then, the device adopts entropy coding to encode the quantized hyper latent \( \hat{v}_t^{(k)} \) and transmits the compressed bitstream to the server. Following [49], [51], we adopt a fully factorized entropy model \( q(\hat{v}_t^{(k)}; \psi^{(k)}) \) to approximate the marginal distribution \( p(\hat{v}_t^{(k)}) \), where the parameters \( \psi^{(k)} \) correspond to the univariate non-parametric density model [49]. From the perspective of data compression, the hyperprior is the side information that allows the server to compress the feature \( \hat{z}_t^{(k)} \) based on a conditional entropy model \( q(\hat{z}_t^{(k)} | \hat{v}_t^{(k)}; \psi^{d}(k)) \). To ensure a good match between the quantized feature \( \hat{z}_t^{(k)} \) and the continuous-valued feature \( \hat{z}_t^{(k)} \), we model each element \( \hat{z}_{t,i}^{(k)} \) in feature \( \hat{z}_t^{(k)} \) as a Gaussian convolved with a unit uniform distribution \( U(-0.5, 0.5) \) as follows:

\[
q(\hat{z}_{t,i}^{(k)} | \hat{v}_{t,i}^{(k)}; \psi^{d}(k)) := \left( \mathcal{N}(\mu_{i}^{(k)}, \sigma_{i}^{(k)}) + U(-0.5, 0.5) \right)(\hat{z}_{t,i}^{(k)}),
\]

with \( \mu_{i}^{(k)}, \sigma_{i}^{(k)} = h_d(\hat{v}_{t,i}^{(k)}; \psi^{d}(k)) \), where the parameters \( \mu_{i}^{(k)} \) and \( \sigma_{i}^{(k)} \) correspond to the \( i \)-th element in \( \mu^{(k)} \) and \( \sigma^{(k)} \), respectively. The function \( h_d(\hat{v}_{t,i}^{(k)}; \psi^{d}(k)) \) is a hyper decoder parameterized by \( \psi^{d}(k) \).
which transforms the quantized latent \( \hat{v}_t^{(k)} \) to the mean and scale parameters \( (\mu^{(k)}, \sigma^{(k)}) \) of the Gaussian distribution. As both the quantized feature \( z_t^{(k)} \) and the latent \( \hat{v}_t^{(k)} \) should be transmitted from the device to the server, the minimum communication cost is the joint entropy \( H(\hat{z}_t^{(k)}, \hat{v}_t^{(k)}) \). With the conditional entropy model \( q(\hat{z}_t^{(k)}|v_t^{(k)}; \psi_d^{(k)}) \) and the factorized entropy model \( q(v_t^{(k)}; \psi_f^{(k)}) \), we formulate a tractable upper bound of the joint entropy \( H(\hat{z}_t^{(k)}, \hat{v}_t^{(k)}) \) as follows:

\[
H(\hat{z}_t^{(k)}, \hat{v}_t^{(k)}) = H(\hat{z}_t^{(k)}|\hat{v}_t^{(k)}) + H(\hat{v}_t^{(k)}),
\]

\[
\leq E[- \log q(\hat{z}_t^{(k)}|v_t^{(k)}; \psi_d^{(k)})]
+ E[- \log q(v_t^{(k)}; \psi_f^{(k)})].
\]  

The first term on the right-hand side of the inequality represents the cost of transmitting \( \hat{z}_t^{(k)} \) by utilizing \( \hat{v}_t^{(k)} \) as the side information. The second term corresponds to the overhead for transmitting the hyper latent \( \hat{v}_t^{(k)} \). To control the communication cost in edge inference, we introduce an extra parameter \( R_{\text{bit}} \) representing the bit constraint. Our objective is to penalize the communication cost when it is larger than the given \( R_{\text{bit}} \). With the above preparation, we formulate a tractable loss function \( L_1 \) in (3), shown at the bottom of the next page. Minimizing it is equivalent to minimizing an upper bound of (1). The details of derivations are deferred to Appendix A.

Remark 1: The parameter \( R_{\text{bit}} \) serves as a constraint on communication overhead. The loss \( L_1 \) will penalize the cost \( \sum_{k=1}^{K} \left[ - \log q(\hat{z}_t^{(k)}|v_t^{(k)}; \psi_d^{(k)}) - \log q(v_t^{(k)}; \psi_f^{(k)}) \right] \) only when it is larger than the given \( R_{\text{bit}} \). Otherwise, the second term of \( L_1 \) will not affect the optimization. Therefore, the training process is less sensitive to the choice of coefficient \( \beta \). In Section III-B, we will develop a temporal entropy model to further minimize the communication overhead by reducing the redundancy among consecutive features.

B. Temporal Entropy Model

Considering the temporal correlation in video frames, it is expected that encoding every extracted feature by using the previous ones as side information achieves a lower bitrate. As shown in Fig. 4, we develop temporal entropy models to exploit the temporal redundancy. Let \( q(\hat{z}_{t-1}^{(k)}; \hat{z}_{t-2}^{(k)}; \psi_p^{(k)}) \) be a variational approximation to the true conditional distribution. We model it using a Gaussian distribution:

\[
q(\hat{z}_{t-1}^{(k)}; \hat{z}_{t-2}^{(k)}; \psi_p^{(k)}) = \mathcal{N}(\mu_t^{(k)}, \sigma_t^{(k)}) * \mathcal{U}(-0.5, 0.5) \left( \hat{z}_{t-1}^{(k)} \right),
\]

where we apply a parametric transform \( \mu^{(k)}, \sigma^{(k)} = h_p(\hat{z}_{t-1}^{(k)}, \hat{z}_{t-2}^{(k)}; \psi_p^{(k)}) \) to predict the mean and variance of the Gaussian distribution. The parameters \( \psi_p^{(k)} \) correspond to the weights of neural network layers. Encoding the features

1With slight abuse of notations, we reuse \( \mu^{(k)}, \sigma^{(k)} \) as the output of the parametric transform.
Algorithm 1 Training Procedures of the Proposed Method

**Input:** Training dataset, initialized parameters $\theta^{(k)}$, $\psi_e^{(k)}$, $\psi_d^{(k)}$, $\psi_f^{(k)}$, $\psi_p^{(k)}$ for $k \in 1:K$, and $\varphi_{0:\tau_1}$, $\phi$.

**Output:** The optimized parameters $\theta^{(k)}$, $\psi_p^{(k)}$ for $k \in 1:K$, and $\phi$.

1: repeat # Train the feature extractors along with hierarchical entropy models and auxiliary predictors.
2: Randomly select a minibatch from the training dataset.
3: while $k = 1$ to $K$ do
4: Extract the features by the feature extractor of device $k$ with parameter $\theta^{(k)}$.
5: Compress the features based on the hierarchical entropy model with parameters $\{\psi_e^{(k)}, \psi_d^{(k)}, \psi_f^{(k)}\}$.
6: end while
7: Output the inference results by the auxiliary predictors with parameters $\varphi_{0:\tau_1}$.
8: Compute the empirical estimation of the loss function $\mathcal{L}_1$ in (3).
9: Update parameters $\theta^{(k)}$, $\psi_e^{(k)}$, $\psi_d^{(k)}$, $\psi_f^{(k)}$ for $k \in 1:K$, and $\varphi_{0:\tau_1}$ through backpropagation.
10: until Convergence of parameters $\theta^{(k)}$, $\psi_e^{(k)}$, $\psi_d^{(k)}$, $\psi_f^{(k)}$ for $k \in 1:K$ and $\varphi_{0:\tau_1}$.
11: repeat # Train the entropy temporal models and the spatial-temporal fusion model in parallel.
12: Randomly select a minibatch from the training dataset.
13: while $k = 1$ to $K$ do
14: Extract the features by the feature extractor of device $k$ with parameter $\theta^{(k)}$.
15: Compress the features based on the temporal entropy model with parameters $\psi_p^{(k)}$.
16: Compute the empirical estimation of the loss function $\mathcal{L}_2^{(k)}$ in (4).
17: Update parameters $\psi_p^{(k)}$ through backpropagation.
18: end while
19: Output the inference results by the spatial-temporal fusion module with parameters $\phi$.
20: Compute the empirical estimation of the loss function $\mathcal{L}_3$ in (4).
21: Update parameters $\phi$ through backpropagation.
22: until Convergence of parameters $\psi_p^{(k)}$ for $k \in 1:K$ and $\phi$.

IV. PERFORMANCE EVALUATION

In this section, we evaluate the performance of the task-oriented communication framework on two edge video analytics tasks, namely, multi-camera pedestrian occupancy prediction and multi-camera object detection.\(^2\)

A. Experimental Setup

1) Tasks and Datasets: We conduct the multi-camera pedestrian occupancy prediction on the Wildtrack dataset [52]. This dataset includes 400 synchronized frames from 7 cameras with overlapping field-of-view, covering a 12 meters by 36 meters ground plane. The objective of this task is to estimate the probabilities of occupancy on the ground plane by exploiting the geometrical constraints from multiple views. For annotation, the ground plane is quantized into a $480 \times 1440$ grid, where each grid cell is a 2.5-centimeter square. Besides, we select the EPFL MVMC Detection dataset [53] to conduct the multi-camera object detection, which contains 6 sequences of frames including 1297 persons, 3553 cars, and 56 buses. The task objective is to assign bounding boxes to describe the spatial location of semantic objects. In the following experiments, we consider that each camera is deployed on

\(^2\)The code is available at https://github.com/shaojiawei07/TOCOM-TEM
an NVIDIA Jetson Orin NX board, and the edge server is equipped with an adequate number of NVIDIA A5000 Tensor Core GPUs.

2) Metric: We mainly investigate the rate-performance tradeoff. The rate refers to the upload overhead, which is calculated by multiplying the average communication cost per frame by the number of cameras. For the multi-camera pedestrian occupancy prediction task, we report the multiple object detection accuracy (MODA) as the performance indicator [54]. This metric accounts for the normalized missed detections and false detections. To evaluate the performance of multi-camera object detection, we adopt the COCO-style mean average precision (mAP)3 to measure the quality of the estimated bounding boxes [55].

3) Proposed Methods: We evaluate the TOCOM-TEM method in the experiments. Particularly, when the parameter $\tau_2$ is set to zero, we adopt the hierarchical entropy model for feature compression, i.e., encoding the current feature independently without leveraging other features as side information. Besides, when setting the parameter $\tau_2 \geq 1$, we adopt the temporal entropy model to compress the extracted features. In the multi-view pedestrian occupancy prediction task, the bit constraint $R_{\text{bit}}$ varies between $10^5$ and $5 \times 10^6$ to control the tradeoff between the communication cost and the performance. For the multi-view object detection task, we set the bit constraint $R_{\text{bit}}$ in the range of $5 \times 10^4$ to $10^6$. The parameter $\beta$ in these two tasks is set to $10^{-5}$.

4) Baselines: In this study, we compare the proposed methods against six different data-oriented communication methods. The deep learning models are trained on the raw images and tested on the compressed data.

- Image compression: We select two conventional image compression methods, JPEG and WebP [56], and two deep image compression (DIC) methods with a factorized prior model [50] and a scale hyperprior model [49] as baselines. The two learning-based methods use neural networks to learn an encoder-decoder pair for image compression and decompression. The encoder outputs a quantized feature as a compact representation of the input image. An entropy model is learned in an end-to-end manner to reduce the bitrate of the quantized feature by entropy coding. In the experiments, the two DIC methods are implemented based on the CompressAI package [57]. The compression ratios achieved by these methods are controlled by adjusting the value assigned to the corresponding quality parameter.

- Video compression: As video frames usually contain abundant amounts of spatial and temporal redundancy, video compression algorithms attempt to store information more compactly. We adopt two well-known video compression algorithms, Advanced Video Coding (AVC) [58] and High Efficiency Video Coding (HEVC) [37], as baselines for comparison. The constant rate factor controls the compression level of video files.

![Fig. 5. The rate-performance curves of different methods for multi-camera pedestrian occupancy prediction task.](image)

We evaluate these baseline methods in terms of the rate-performance curve by setting different quality values.

5) Neural Network Architecture: In the multi-camera pedestrian occupancy prediction task, we adopt the MVDet model [47] as the backbone to estimate the pedestrian position by incorporating multiple camera views. The MVDet model is composed of two main components: a feature extractor and a spatial aggregator. The feature extractor is similar to ResNet-18 [16], but the last two layers are removed and the last three strided convolutions are replaced by dilated convolutions. The spatial aggregator consists of three convolutional layers to exploit the spatial neighbor information of multiple views. Besides, the hierarchical entropy model adopts five convolutional layers and five deconvolutional layers as a hyper encoder and a hyper decoder, respectively. The temporal entropy model employs three convolutional layers as the parametric transform.

For the multi-camera detection task, we select the fully convolutional one-stage object detector (FCOS) [59] to solve object detection in an anchor-free manner. The FCOS framework utilizes a ResNet-50 model and a feature pyramid network (FPN) [60] for feature extraction. An anchor-free detector outputs the bounding boxes based on the multi-level features. Besides, we construct a residual building block consisting of three convolutional layers to fuse the feature maps from different cameras to exploit the spatial correlation. To effectively compress the multi-scale features, the hierarchical entropy model contains four convolutional layers and four deconvolutional layers, and the temporal entropy model consists of five convolutional layers.

B. Multi-Camera Pedestrian Occupancy Prediction

This task aims at incorporating multiple camera views to predict pedestrian occupancy. The multi-camera system alleviates the impact of occlusions in crowded scenes, but it induces much higher communication overhead compared with the single-camera system. Fig. 5 presents the rate-performance curves of different methods for multi-camera pedestrian occupancy prediction.
occupancy prediction, where the horizontal axis shows the communication cost, and the vertical axis corresponds to the inference performance. For data-oriented communication, the result shows that video compression methods, i.e., AVC and HEVC, consistently outperform the image compression methods. This is because video compression methods can reduce the redundancy among consecutive frames to achieve a higher compression ratio. Among all the configurations in the figure, our TOCOM-TEM method performs the best. This demonstrates the effectiveness of the task-oriented communication framework that discards irrelevant information while maintaining inference performance. Particularly, setting the parameters $\tau_1, \tau_2$ to 1 in TOCOM-TEM achieves a better rate-performance tradeoff compared with setting them to 0. This is attributed to the ability of the entropy model and spatial-temporal fusion module to exploit the temporal correlation among features. When encoding feature $\hat{z}_t^{(k)}$, the temporal entropy model adopts the previous feature $\hat{z}_{t-1}^{(k)}$ as side information to reduce the bitrate. Besides, the spatial-temporal fusion module reduces the uncertainty in prediction by jointly utilizing the current received features $\hat{z}_t^{(1:K)}$ and the previous features $\hat{z}_{t-1}^{(1:K)}$.

Next, we conduct a comparison of the latency across different methods, where the overall latency in edge inference comprises on-device computation time, transmission latency, and server-side computation time. In order to prevent over-compression of input data, we adjust the compression ratio of each method to ensure that the MODA score is approximately 87%. Fig. 6 depicts the inference latency at various communication rates. In many cases, the transmission process is the main bottleneck causing delays in edge inference, especially when the communication rate is low. The extra feature extraction step in our method increases the complexity on the device side, but it effectively removes the task-irrelevant information and largely reduces the communication overhead. Therefore, the proposed TOCOM-TEM method consistently exhibits the smallest latency at various communication rates.
C. Multi-Camera Object Detection

The object detection task in a multi-camera environment enjoys the benefit of the images taken from different angles, which helps to take into account occlusions between objects for better bounding box regression. We compare the rate-performance tradeoff of different methods, and the results are shown in Fig. 7. The data-oriented communication methods (i.e., image compression and video compression) lead to higher communication cost compared with the task-oriented communication method. This is attributed to the fact that data-oriented communication simply guarantees the correct reception of every single transmitted bit without optimizing the communication strategy for the downstream task, so a large amount of the transmitted data may be irrelevant to the inference task. In our TOCOM-TEM method, using the temporal entropy model (τ2 = 1) achieves a better rate-performance tradeoff compared with the hierarchical entropy model (τ2 = 0). This is because using the previous features as the side information in entropy coding further reduces the encoded data size. Besides, Fig. 8 displays the overall latency of various methods that achieve approximately 60% mAP. While our method introduces additional complexity on the device side due to the complex feature extraction process, the proposed TOCOM-TEM method still enables low-latency inference. This is because the increase in computation latency is significantly outweighed by the reduction in transmission latency.

D. Ablation Study

In this subsection, we evaluate the impact of the feature fusion module and the parameters τ1, τ2 of the proposed TOCOM-TEM method on the edge video analytics systems. Besides, we discuss the bit allocation of the transmitted features in data-oriented communication and task-oriented communication.

1) Impact of the Feature Fusion Module: To evaluate the effectiveness of the feature fusion module in our method, we consider a baseline denoted as (TOCOM-TEM w/o feature fusion), which does not exploit the correlation among multi-camera features in the inference and is in essence a method for single-camera inference. We compare the rate-performance curves of the TOCOM-TEM method with and without the feature fusion module in the multi-camera object detection task. In this experiment, the parameters τ1 and τ2 are set to 0 and 1, respectively. As shown in Fig. 9, the proposed TOCOM-TEM method achieves a better rate-performance tradeoff. This is due to the feature fusion module that improves performance under occlusion by leveraging the spatial dependency among multiple cameras.

2) Impact of the Parameter τ1: We investigate the performance of the TOCOM-TEM method with different values of τ1 in the multi-camera pedestrian occupancy prediction task. The bit constraint Rbit is set to 2.5 × 10^5 in this experiment. As shown in Fig. 10, increasing the value of the parameter τ1 improves the MODA score. This result is consistent with the analysis in Section II-C.3 that leveraging both the current received features as well as τ1 previous features reduces the uncertainty in the prediction.

3) Impact of the Parameter τ2: To investigate the performance of the temporal entropy model in the proposed task-oriented communication framework, we evaluate the communication cost of our method by varying the value of parameter τ2. Particularly, we set the bit constraint Rbit to 10^6 and 2 × 10^5 in the multi-camera pedestrian occupancy task and the multi-camera object detection task, respectively. The temporal entropy model encodes the current feature Zt(k) conditional on the previous τ2 features {Zt−1(k), Zt−2(k)}. As shown in Fig. 11, increasing the value of τ2 leads to a reduction in communication costs. These results align with the information theory that the conditional entropy H(Zt(k)|Zt−1(k), . . . , Zt−τ2(k)) decreases monotonically as τ2 increases.

4) Bit Allocation in Feature Transmission: This part discusses the difference between the extracted features in the DIC (scale hyperprior model) method and our TOCOM-TEM method. To achieve the same MODA of around 85% in the multi-camera pedestrian occupancy task, the DIC method costs around 130 KB in communication overhead. In comparison, the communication cost of our TOCOM-TEM method with parameter τ2 = 0 and τ2 = 1 are 50 KB and 20 KB, respectively. Fig. 12 shows visualizations of the normalized bitrate allocations for the input frames. The second row of the figure corresponds to the extracted features of the DIC method. It appears that its entropy model allocates many bits to some unique areas in an image to preserve all its characteristics in the reconstructed image. However, this data-oriented communication scheme leads to high communication overhead, and a large part of the information is irrelevant to the downstream task. On the contrary, the features of our method shown in the third and fourth rows mainly focus on task-relevant areas.

![Graphs showing communication cost comparison](image-url)
(e.g., the foot points of pedestrians) that are useful to predict occupancy. This is because the task-oriented communication framework is optimized for a specific task, rather than for reliable video transmission. Besides, setting $\tau_2 = 1$ allows the temporal entropy model to utilize the previous feature as side information in entropy coding, which further reduces the communication overhead.

V. CONCLUSION

This paper investigated task-oriented communication for device edge analytics, where multiple devices transmit the task-relevant features to an edge server for inference. Our proposed communication framework is built upon the deterministic IB principle and the temporal entropy model for feature extraction and encoding, respectively. Besides, a spatial-temporal fusion module was proposed to integrate both the current received features and the previous features at the server to improve the performance of the edge video analytics system. Compared with the data-oriented communication strategies, our proposed TOCOM-TEM method consistently outperforms the baselines in terms of the rate-performance tradeoff.

This study demonstrated the effectiveness of task-oriented communication for edge video analytics, which serves for video analytics applications rather than for video reconstruction. Besides, the temporal entropy model proposed in this work showed the potential to reduce the communication overhead by exploiting the correlation among features of consecutive frames. For future research, it is interesting to further exploit the spatial-temporal correlation of multiple devices in edge video analytics systems, and formalize the optimal rate-performance tradeoff in task-oriented communication. This would have a broad impact on the intelligent video analytics applications that necessitate cooperation among multiple cameras.

APPENDIX A

DERIVATION OF THE UPPER BOUND OF (1)

Recall that the objective function in (1) includes the term 

$$-I(Y_t, \ldots, Y_{t+\tau_1}; Z_t^{(1:K)}) = H(Y_t, \ldots, Y_{t+\tau_1}|\hat{Z}_t^{(1:K)}) - H(Y_t, \ldots, Y_{t+\tau_1})$. Writing the conditional entropy with the variational distribution, it becomes:

$$H(Y_t, \ldots, Y_{t+\tau_1}|\hat{Z}_t^{(1:K)}) = \mathbb{E}\left\{ -\log q(y_y, \ldots, y_{t+\tau_1}|\hat{z}_t^{(1:K)}; \varphi_0; \tau_1) \right\}$$

$$- \mathbb{E}\left\{ -\log \frac{p(y_y, \ldots, y_{t+\tau_1}|\hat{z}_t^{(1:K)})}{q(y_y, \ldots, y_{t+\tau_1}|\hat{z}_t^{(1:K)}; \varphi_0; \tau_1)} \right\}.$$

As the KL-divergence is not negative, we have

$$H(Y_t, \ldots, Y_{t+\tau_1}|\hat{Z}_t^{(1:K)}) \leq \mathbb{E}\left\{ -\log q(y_y, \ldots, y_{t+\tau_1}|\hat{z}_t^{(1:K)}) \right\}. \quad (5)$$

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
As the joint entropy $H(Y_1, \ldots, Y_{t+\tau})$ is a constant, minimizing $\mathbb{E}\{-\log q(y_1, \ldots, y_{t+\tau}; z_{t}^{1:K})\}$ is equivalent to minimizing an upper bound of $-I(Y_1, \ldots, Y_{t+\tau}; \hat{Z}_t^{1:K})$.

The entropy $H(\hat{Z}_t^{(k)})$ in (1) represents the communication cost. In the proposed method, we introduce a latent variable $\hat{v}_t^{(k)}$ as side information to encode $\hat{z}_t^{(k)}$. Both the quantized feature $\hat{z}_t^{(k)}$ and the latent $\hat{v}_t^{(k)}$ should be transmitted from the device to the server. Therefore, the minimum communication cost is the joint entropy $H(\hat{z}_t^{(k)}, \hat{v}_t^{(k)})$. Write the joint entropy term in full with the variational distributions $q(\hat{z}_t^{(k)} | \hat{v}_t^{(k)}; \psi_d^{(k)})$ and $q(\hat{v}_t^{(k)}; \psi_p^{(k)})$. It is as follows:

$$H(\hat{z}_t^{(k)}, \hat{v}_t^{(k)}) = \mathbb{E}\{-\log q(\hat{z}_t^{(k)} | \hat{v}_t^{(k)}; \psi_d^{(k)})q(\hat{v}_t^{(k)}; \psi_p^{(k)})\} - \mathbb{E}\{-\log q(\hat{z}_t^{(k)} | \hat{v}_t^{(k)}; \psi_d^{(k)})q(\hat{v}_t^{(k)}; \psi_p^{(k)})\}.$$  

As the KL-divergence is larger than zero, we have

$$H(\hat{z}_t^{(k)}, \hat{v}_t^{(k)}) \leq \mathbb{E}\{-\log q(\hat{z}_t^{(k)} | \hat{v}_t^{(k)}; \psi_d^{(k)})q(\hat{v}_t^{(k)}; \psi_p^{(k)})\}.$$  

(6)

By combining (5) and (6), we can find that minimizing the $L_1$ loss in (3) is equivalent to minimizing an upper bound of (1).

REFERENCES

[1] M.-C. Chang et al., “AI city challenge 2020—Computer vision for smart transportation applications,” in Proc. Conf. Comput. Vis. Pattern Recognit. Workshop, Jun. 2020, pp. 2638–2647. [Online]. Available: https://ieeexplore.ieee.org/document/9151055

[2] N. A. Othman and I. Aydin, “A new IoT combined body detection of people by using computer vision for security application,” in Proc. 9th Int. Conf. Comput. Commun. Netw. (CICCN), Ercan, North Cyprus, Sep. 2017, pp. 108–112.

[3] J. Gao, Y. Yang, P. Lin, and D. S. Park, “Computer vision in healthcare applications,” J. Healthc. Eng., vol. 2018, pp. 1–4, Mar. 2017.

[4] G. Ananthanarayanan et al., “Real-time video analytics: The killer app for edge computing,” Computer, vol. 50, no. 10, pp. 58–67, 2017.

[5] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang, “Edge intelligence: Paving the last mile of artificial intelligence with edge computing,” Proc. IEEE, vol. 107, no. 8, pp. 1738–1762, Aug. 2019.

[6] J. Shao, Y. Mao, and J. Zhang, “Task-oriented communication for multi-device cooperative edge inference,” IEEE Trans. Wireless Commun., vol. 22, no. 1, pp. 73–87, Jan. 2023.

[7] J. Shao, Y. Mao, and J. Zhang, “Learning task-oriented communication for edge inference: An information bottleneck approach,” IEEE J. Sel. Areas Commun., vol. 40, no. 1, pp. 197–211, Jan. 2022.

[8] O. A. Hanna, Y. H. Ezzeildin, T. Sadjadpour, C. Fragouli, and S. Diggavi, “On distributed quantization for classification,” IEEE J. Sel. Areas Inf. Theory, vol. 1, no. 1, pp. 237–249, May 2020.

[9] S. Singhal, V. Raghunathan, and A. Raghunathan, “Communication-efficient view-pooling for distributed multi-view neural networks,” in Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE), Grenoble, France, Mar. 2020, pp. 1390–1395.

[10] J. Choi, Z. Hakimi, P. W. Shin, J. Sampson, and V. Narayanan, “Context-aware convolutional neural network over distributed system in collaborative computing,” in Proc. ACM/IEEE Design Automat. Conf. (DAC), Las Vegas, NV, USA, Jun. 2019, pp. 1–6.

[11] Y. Zhou, J. Xiao, Y. Zhou, and G. Loiano, “Multirobot collaborative perception with graph neural networks,” IEEE Robot. Autom. Lett., vol. 7, no. 2, pp. 2289–2296, Apr. 2022.

[12] S. Xie, S. Ma, M. Ding, Y. Shi, M. Tang, and Y. Wu, “Robust information bottleneck for task-oriented communication with digital modulation,” 2022, arXiv:2209.10382.

[13] N. Tishby, F. C. Pereira, and W. Bialek, “The information bottleneck method,” in Proc. Annu. Allerton Conf. Commun. Control Comput., Monticello, IL, USA, Oct. 2000, pp. 1–11.

[14] M. Sanderme r, R. Schltter, and H. N ey, “LSTM neural networks for language modeling,” in Proc. Annu. Int. Conf. Int. Speech Commun. Assoc., Portland, OR, USA, Sep. 2012, pp. 194–197.

[15] A. Vaswani et al., “Attention is all you need,” in Proc. Adv. Neural Inf. Process. Syst., vol. 30, Dec. 2017, pp. 1–11.

[16] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Las Vegas, NV, USA, Jan. 2016, pp. 770–778.

[17] E. Ristani and C. Tomasi, “Features for multi-target multi-camera tracking and re-identification,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Salt Lake City, UT, USA, Jun. 2018, pp. 6036–6046.

[18] H. Liu, Y. Tian, Y. Wang, L. Pang, and T. Huang, “Deep relative distance learning: Tell the difference between similar vehicles,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Las Vegas, NV, USA, Jun. 2016, pp. 2167–2175.

[19] G. Farnebäck, “Two-frame motion estimation based on polynomial expansion,” in Proc. Scandinavian Conf. Image Anal., Halmstad, Sweden, Jul. 2003, pp. 363–370.

[20] D. Stroise and D. J. Schwab, “The deterministic information bottleneck,” Neural Comput., vol. 11, no. 6, pp. 1611–1630, Jun. 2017.

[21] E. Calvanese Strinati and S. Barbarossa, “6G networks: Beyond Shannon towards semantic and goal-oriented communications,” Computer, vol. 50, no. 1, pp. 19–25, Jan. 2020.

[22] J. Shao and J. Zhang, “Communication-computation trade-off in resource-constrained edge inference,” IEEE Commun. Mag., vol. 58, no. 12, pp. 20–26, Dec. 2020.

[23] A. Erfan Eshratifar, A. Esmaili, and M. Pedram, “BottleNet: A deep bottleneck for task-oriented communication with out-of-band transmission for scene classification in unmanned aerial systems,” in Proc. IEEE Int. Conf. Commun., Hong Kong, China, Jun. 2022, pp. 1302–1307.

[24] A. Vaswani et al., “Attention is all you need,” in Proc. Adv. Neural Inf. Process. Syst., vol. 30, Dec. 2017, pp. 1–11.

[25] Y. Dubois, B. Bloom-Reddy, K. Ulrich, and C. J. Maddison, “Lossy compression for lossless prediction,” in Proc. Int. Conf. Learn. Represent. Workshop Neural Comput., May 2021, pp. 1–15. [Online]. Available: https://openreview.net/forum?id=GHCs5NhoR8Q

[26] H. Li et al., “Task-oriented communication with out-of-distribution detection: An information bottleneck framework,” 2023, arXiv:2305.12423.

[27] A. Erfan Eshratifar, A. Esmaili, and M. Pedram, “BottleNet: A deep learning architecture for intelligent mobile cloud computing services,” 2019, arXiv:1902.01000.

[28] M. Jankowski, D. Gündüz, and K. Mikolajczyk, “Wireless image retrieval at the edge,” IEEE J. Sel. Areas Commun., vol. 39, no. 1, pp. 89–100, Jan. 2021.

[29] J. Shao, H. Zhang, T. Mao, and J. Zhang, “Branchy-GNN: A device-edge co-inference system for high efficiency video coding (HEVC) standard,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), Canada, Jun. 2021, pp. 8488–8492.

[30] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, “Overview of the H.264/AVC video coding standard,” IEEE Trans. Circuits Syst. Video Technol., vol. 13, no. 7, pp. 560–576, Jul. 2003.

[31] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, “Overview of the high efficiency video coding (HEVC) standard,” IEEE Trans. Circuits Syst. Video Technol., vol. 22, no. 12, pp. 1649–1668, Dec. 2012.
[38] G. Lu, W. Ouyang, D. Xu, X. Zhang, C. Cai, and Z. Gao, “DVC: An end-to-end deep video compression framework,” in Proc. Conf. Comput. Vis. Pattern Recognit., Georgia Tech, GA, USA, Jun. 2019, pp. 11006–11015.

[39] J. Liu et al., “Conditional entropy coding for efficient video compression,” in Proc. Eur. Conf. Comput. Vis., Aug. 2020, pp. 453–468. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-030-58520-4_27

[40] T.-Y. Tung and D. Gündüz, “DeepViVe: Deep learning-aided wireless video transmission,” IEEE J. Sel. Areas Commun., vol. 40, no. 9, pp. 2570–2583, Sep. 2022.

[41] D. Liu, Y. Li, J. Lin, H. Li, and F. Wu, “Deep learning-based video coding: A review and a case study,” ACM Comput. Surv., vol. 53, no. 1, pp. 1–35, 2020.

[42] S. Ma, X. Zhang, C. Jia, Z. Zhao, S. Wang, and S. Wang, “Image and video compression with neural networks: A review,” IEEE Trans. Circuits Syst. Video Technol., vol. 30, no. 6, pp. 1683–1698, Apr. 2019.

[43] E. Agustsson, D. Minnen, N. Johnston, J. Balle, S. J. Hwang, and G. Toderici, “Scale-space flow for end-to-end optimized video compression,” in Proc. Conf. Comput. Vis. Pattern Recognit., Jun. 2020, pp. 8555–8564. [Online]. Available: https://ieeexplore.ieee.org/document/9157366

[44] Z. Hu, G. Lu, and D. Xu, “FVC: A new framework towards deep video compression in feature space,” in Proc. Conf. Comput. Vis. Pattern Recognit., Jun. 2021, pp. 1502–1511.

[45] J. Li, B. Li, and Y. Lu, “Deep contextual video compression,” in Proc. Adv. Neural Inf. Process. Syst., Dec. 2021, pp. 1–12. [Online]. Available: https://openreview.net/forum?id=evqzNxmXsl3

[46] E. Ristani, F. Solera, R. Zou, R. Cucchiara, and C. Tomasi, “Performance measures and a data set for multi-target, multi-camera tracking,” in Proc. Eur. Conf. Comput. Vis., Amsterdam, The Netherlands, Oct. 2016, pp. 17–35.

[47] Y. Hou, L. Zheng, and S. Gould, “Multiview detection with feature perspective transformation,” in Proc. Eur. Conf. Comput. Vis., Aug. 2020, pp. 1–18. [Online]. Available: https://www.ecva.net/papers/eccv_2020/132520001.pdf

[48] K. B. Letaief, Y. Shi, J. Lu, and J. Lu, “Edge artificial intelligence for 6G: Vision, enabling technologies, and applications,” IEEE J. Sel. Areas Commun., vol. 40, no. 1, pp. 5–36, Nov. 2021.

[49] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, “Variational image compression with a scale hyperprior,” in Proc. Int. Conf. Learn. Represent., Vancouver, BC, Canada, 2018, pp. 1–23.

[50] J. Ballé, V. Laparra, and E. Simoncelli, “End-to-end optimized image compression,” in Proc. Int. Conf. Learn. Represent., Toulon, France, Apr. 2017, pp. 1–27.

[51] D. Minnen, J. Ballé, and G. D. Toderici, “Joint autoregressive and hierarchical priors for learned image compression,” in Proc. Neural Inf. Process. Syst., Montreal, QC, Canada, Dec. 2018, pp. 1–10.

[52] T. Chavdarova et al., “WILDBACK: A multi-camera HD dataset for dense unscripted pedestrian detection,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Salt Lake City, UT, USA, Jun. 2018, pp. 5030–5039.

[53] G. Roig, X. Boix, H. Ben Shitrit, and P. Fua, “Conditional random fields for multi-camera object detection,” in Proc. Int. Conf. Comput. Vis., Barcelona, Spain, Nov. 2011, pp. 563–570.

[54] R. Kasturi et al., “Framework for performance evaluation of face, text, and vehicle detection and tracking in video,” Data, metrics, and protocol,” IEEE Trans. Pattern Anal. Mach. Intel., vol. 31, no. 2, pp. 319–336, Mar. 2008.

[55] T.-Y. Lin et al., “Microsoft COCO: Common objects in context,” in Proc. Eur. Conf. Comput. Vision, Zurich, Switzerland, Sep. 2014, pp. 740–755.

[56] Z. Si and K. Shen, “Research on the WebP image format,” in Proc. Adv. Graph. Commun. Packag. Technol. Mater., Dec. 2015, pp. 271–277.

[57] J. Bégaud, E. Racapé, S. Feltman, and A. Pushparaja, “CompressAI: A PyTorch library and evaluation platform for end-to-end compression research,” 2020, arXiv:2011.03029.

[58] I. E. Richardson, The H. 264 Advanced Video Compression Standard. Hoboken, NJ, USA: Wiley, 2011.

[59] Z. Tian, C. Shen, H. Chen, and T. He, “FCOS: Fully convolutional onestage object detection,” in Proc. Int. Conf. Comput. Vis., Nov. 2019, pp. 9627–9636.

[60] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” in Proc. Conf. Comput. Vis. Pattern Recognit., Honolulu, HI, USA, Jul. 2017, pp. 2117–2125.

Jiawei Hao (Graduate Student Member, IEEE) received the B.Eng. degree in telecommunication engineering from the Beijing University of Posts and Telecommunications, Beijing, China, in 2019. He is currently pursuing the Ph.D. degree with The Hong Kong University of Science and Technology. His current research interests include edge artificial intelligence, task-oriented communications, neural compression, and federated learning.

Xinjie Zhang (Graduate Student Member, IEEE) received the B.Eng. degree from the School of Electronic Information and Engineering, South China University of Technology, Guangzhou, China. He is currently pursuing the Ph.D. degree with The Hong Kong University of Science and Technology, Hong Kong. His current research interests include neural compression, implicit neural representation, and edge artificial intelligence.

Jun Zhang (Fellow, IEEE) received the B.Eng. degree in electronic engineering from the University of Science and Technology of China in 2004, the M.Phil. degree in information engineering from The Chinese University of Hong Kong in 2006, and the Ph.D. degree in electrical and computer engineering from The University of Texas at Austin in 2009.

He is currently an Associate Professor with the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology. He has coauthored the book Fundamentals of LTE (Prentice-Hall, 2010). His current research interests include wireless communications and networking, mobile edge computing and edge AI, and cooperative AI. He is an IEEE ComSoc Distinguished Lecturer. He was a co-recipient of several best paper awards, including the 2021 Best Survey Paper Award of the IEEE Communications Society, the 2019 IEEE Communications Society and Information Theory Society Joint Paper Award, and the 2016 Marconi Prize Paper Award in Wireless Communications. Two papers he co-authored received the Young Author Best Paper Award of the IEEE Signal Processing Society in 2016 and 2018. He served as the MAC Track Co-Chair for the IEEE Wireless Communications and Networking Conference (WCNC) in 2011 and the Co-Chair for the Wireless Communications Symposium of the IEEE International Conference on Communications (ICC) in 2021. He also received the 2016 IEEE ComSoc Asia-Pacific Best Young Researcher Award. He was an Editor of IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS from 2015 to 2020. He is an Editor of IEEE TRANSACTIONS ON COMMUNICATIONS and IEEE TRANSACTIONS ON MACHINE LEARNING IN COMMUNICATIONS AND NETWORKING.