Vidfra: Collaborative Real-time Video Understanding over Dynamic Networks

***Abstract*—Video understanding, integral to a wide range of real-world applications, requires the inference of a sequence of temporal video frames. However, existing video analytics approaches are primarily developed for basic frame-level tasks based on single frames. These approaches ignore the temporal dependency and semantic relevance of video frames, failing to perform complex video understanding tasks in dynamic networks. This paper presents Vidfra, a novel edge-to-cloud framework for growing video understanding tasks. Vidfra operates by intelligently selecting the most representative video partition, adjusting the video downsampling rate on the edge side, and subsequently reconstructing the original temporal sequence via cloud-side analytics-aware video interpolations. Vidfra judiciously leverages the temporal-level redundancy among video frames to reduce bandwidth consumption for better analytics results. Comprehensive experiments conducted on real-world video understanding tasks reveal that Vidfra consistently achieves an average bandwidth reduction of approximately 57.7% and an average end-to-end latency reduction of 52.8%, while maintaining a relative accuracy exceeding 91.1% compared to analytics results obtained from the original videos.**

***Index Terms*—Video understanding, edge-cloud collaboration, video frame interpolation, video streaming**

I. Introduction

Driven by the advancements in artificial intelligence and the wide deployment of networked cameras, the landscape of video analytics is rapidly evolving. The integration of sophisticated AI models has enabled the extraction of deeper insights from video data, going beyond traditional object recognition to understand intricate behaviors, interactions, and anomalies. Unlike the classical video analytics tasks that operate spatial analysis on distinct frames, complex sequence-level video analytics tasks, referred as video understanding, require a comprehensive spatial-temporal understanding of a sequence of temporal video frames. Video understanding tasks such as video action recognition (VAR), video speech recognition (VSR, namely lip reading) and video question answering (VQA) have wide applications in traffic surveillance [1], human-computer interaction [2], mass video retrieval [3], and others.

Video understanding. The divergence between video understanding tasks and traditional video analytics [4] primarily stems from the treatment of temporal information. As illustrated in Fig. 1, classical vision tasks in video analytics, such as object detection or classification, typically operate on static frames and do not account for the temporal continuity between frames. This approach is sufficient for identifying individual items within a single frame but fails to capture the dynamics and interactions over time, which are essential for understanding complex sequences like human activities or interactive events.

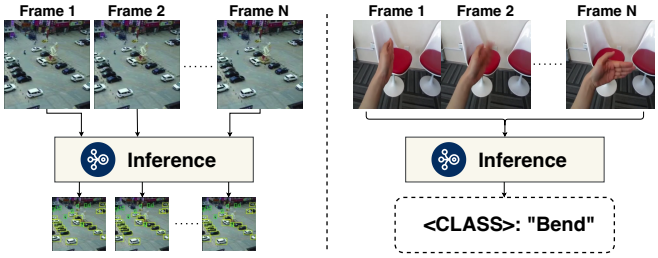


Fig. 1. Comparison between traditional video analytics pipeline (left) and video understanding inference pipeline (right). Traditional video analytics mainly focuses on frame-level analytics tasks (e.g. object detection), which generate inference result for each individual video frames. Video understanding are sequence-level analytics tasks (i.e. action recognition), which requires a sequence of video frames to generate a result.

Video understanding tasks address this limitation by analyzing the entire sequence of frames, thus requiring sophisticated methodologies that can interpret both spatial and temporal data simultaneously.

To address these complex requirements, specialized deep neural networks such as 3D Convolutional Neural Networks (3DCNNs) [5], Temporal Segment Networks (TSNs) [6] and video transformers [7]–[10], have been developed. 3DCNNs incorporate time as a dimension alongside space, allowing for the direct capture of temporal patterns. TSNs, on the other hand, divide videos into segments to process long-range temporal dependencies, enabling a comprehensive analysis that accommodates variations in video content and length. This design shift is crucial for capturing dynamics and temporal relationships within video sequences, which are essential for understanding actions, events, and behaviors over time.

Challenges. The impact of video understanding tasks on video analytics systems has rarely been explored and such tasks raise several unique technical challenges in dynamic networks. Unlike traditional frame-level video analytics tasks, which analyze single frames independently, video understanding relies on the temporal continuity of frames. Latency spikes can cause gaps or delays between frames, impairing the ability to track and understand actions or events that unfold over time, leading to delayed or incorrect inferences. Additionally, dynamic networks are prone to packet loss, which can result in missing or corrupted frames. While the loss of a few frames might not drastically affect frame-level tasks, for video understanding, where each frame contributes to the overall context, even minor losses can degrade performance. Packet loss disrupts the sequential analysis, causing errors to propagate

through the video sequence and resulting in compounded inaccuracies.

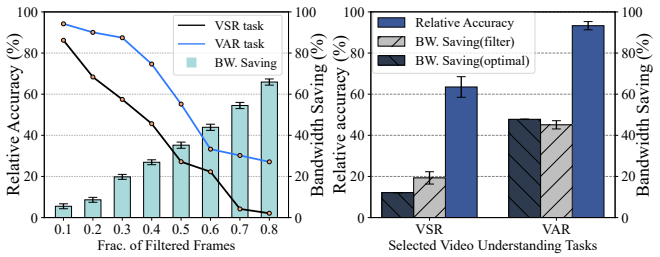
However, when we dive into it, we find that existing techniques for efficient video analytics, such as frame filtering, region of interest (RoI) extraction, frame compression, and adaptive configuration [11]–[16] are primarily cater to basic video analytics tasks that operate on single frames. They cannot help, mostly can even harm, the successful running of video understanding tasks. For instance, frame filtering designed for frame-level video analytics tasks can lead to a severe loss of a video's temporal information, which is crucial for video understanding. RoI extraction is not universally applicable across all video understanding tasks. Meanwhile, adaptive configuration may disrupt the temporal information in videos, leading to a significant drop in accuracy. *All of these limitations call for a rethinking of how video data should be processed, streamed, and analyzed in video analytics systems to better serve video understanding tasks.*

Goal. We study an edge-cloud collaborative video understanding system where videos are captured in real-time on edge devices and continuously streamed to cloud servers over dynamic network conditions for real-time analysis. The overarching goal of our work with Vidfra is to enhance the efficiency and effectiveness of real-time video understanding systems, particularly in dynamic network environments. We aim to address specific challenges associated with video analytics, focusing on three main objectives: (1) Reduce bandwidth consumption, (2) Decrease end-to-end latency, (3) Maintain or improve analytical accuracy. Bandwidth consumption and end-to-end latency reflect the efficiency while the analytical accuracy represents the effectiveness of our system.

Vidfra. Vidfra optimizes video understanding tasks through a sophisticated framework designed around three integral components:

- Edge-Side Video Processing: Videos are initially processed at the edge by a context-aware video scorer that evaluates spatial-temporal relevance, identifying segments with task-relevant events. These segments are selectively sampled at adaptive rates determined by a cloud-based profiler, effectively minimizing the data transmitted while preserving crucial temporal details.
- Cloud-Side Video Enhancement: In the cloud, a video frames interpolation module reconstructs the temporal sequences of downsampled videos. An analytics-aware process to ensure the restoration of essential temporal information for precise video analysis.
- Adaptive Streaming and Configuration Profiling: Vidfra includes an adaptive streaming controller that dynamically adjusts streaming parameters like bitrate based on real-time network conditions, optimized using a sparse-reward algorithm. The profiler also fine-tunes sampling and interpolation rates, balancing analytical accuracy with system efficiency.

To the best of our knowledge, Vidfra is the first video analytics framework for efficient video understanding.



(a) Area Diff. Filter. (b) Neural-network Classifier.

Fig. 2. Limitations of existing video analytics pipeline designs on various video understanding tasks: (a) analytical accuracy and bandwidth saving ratio across various filtered frame fractions by using frame filter based on low-level vision feature, (b) analytical accuracy and bandwidth saving ratio by using binary neural network classifier, (c) extra latency introduced by unique RoI encoding process over various tasks.

In summary, this paper makes several significant contributions to the field of video analytics for video understanding tasks. We first comprehensively examine the applicability of existing video analytics techniques and expose their limitations in terms of performance, efficiency, and generality. We introduce Vidfra, a novel edge-cloud collaborative video analytics framework specifically designed for video understanding. Vidfra effectively exploits temporal redundancy and leverages additional cloud-side computation to achieve a substantial reduction in network bandwidth consumption while maintaining high inference accuracy. Additionally, Vidfra introduces innovative approaches in the pre-processing, streaming, and post-processing modules of video analytics systems, including a novel contextual-aware frame selection algorithm, a reinforcement learning-based adaptive streaming algorithm with sparse rewards, and an analytics-aware frame interpolation algorithm, all tailored for video understanding tasks. Our extensive experiments across a wide range of video understanding tasks and datasets demonstrate that Vidfra significantly reduces bandwidth consumption by 57.7%, maintains a relative analytic accuracy of over 91.1% compared to unprocessed videos, and reduces latency in video analytics by 52.8%.

The remainder of this paper is organized as follows. We first introduce our measurement studies in Section II to identify the limitations and challenges of existing video analytics pipelines when applied to video understanding. We then illustrate Vidfra's detailed design in Section III and evaluate its performance in Section IV. Related work is discussed in Section V, followed by the conclusion in Section VI.

II. Motivation

Many existing video analytics pipelines focus on balancing communication costs and analytical accuracy using frame filtering, RoI encoding, adaptive bitrate streaming, and so on. However, these techniques are primarily tailored for classical vision tasks and do not adequately address the needs of sequence-level video analytics tasks. We reveal this mismatch and its consequences in detail as follows.

Existing frame filter techniques hurt the temporal information. We notice that a significant portion of raw video data typically encompasses redundant temporal information. Only a limited percentage of frames in each video contain pertinent task-relevant events, which indicates a large optimization room to reduce the redundancy such as unrelated frames or events. We analyze the temporal redundancy over various video understanding datasets. For instance, in the Egogesture [17] dataset for video action recognition, a fraction under 45.0% of frames hold ground truth, and similarly, only around 53.2% frames contain ground truth information in the TGIF-QA dataset [18] for video question answering. Notably, omitting unrelated video frames results in minimal fluctuations in analytics accuracy, typically less than 5%. *This observation underscores a significant opportunity for optimization in the temporal domain of video understanding tasks, where strategic frame sampling can greatly improve efficiency without compromising analytical effectiveness.*

However, existing approaches of preprocessing videos on the edge via frame filtering show severe limitations in performing video understanding tasks. One typical method [12] employs low-level vision features, such as area, pixel, edge, and histogram of oriented gradients (HOG) features. These features are inefficient in capturing temporal information, which is crucial for video understanding tasks. We conducted a measurement study to evaluate the effectiveness of using various visual features as a guide for frame filtering. This process involves calculating the differences between adjacent frames and subsequently filtering out frames with relatively low differences. We evaluated the relative accuracy and bandwidth savings ratio across a range of filtered frame fractions, from 0.1 to 0.8, employing features such as area, pixel, edge, and HOG. Considering the space limits and to illustrate our findings, we present the results obtained using the area feature as an example in Figure 2(a). The observation still holds for other features mentioned above. Our analysis reveals a substantial accuracy drop with even a modest filter fraction up to 0.3 in VSR and 0.4 in VAR, leading to a modest bandwidth saving of around 19.8% and 26.9% respectively. Achieving a 90% relative accuracy allows for saving only 5.5% and 9.3% of bandwidth, respectively, indicating inadequate efficiency.

Alternative prevalent approaches [11], [19] utilize lightweight neural network classifiers at the edge, leveraging feature maps from compact neural networks such as Tiny-yolo [20] or other computer vision models for guidance. These classifiers determine frame retention based on the detection of task-specific objects of interest within the frames. However, the mere presence of task-related objects does not inherently ensure the accurate localization of the underlying truth events, as such methods primarily capture instance-level spatial information, largely omitting the temporal aspect. While this strategy may prove beneficial for certain applications where the presence of specific objects aligns with the ground truth events, such as the specific dataset we adopt for VAR task, it falls short for scenarios demanding more intricate temporal data, like VSR, video question answering(VQA), and video captioning. Our evaluation of this approach, which applies Tiny-yolo as a frame classifier for streaming, is depicted in Figure 2(b). It is evident that for multifaceted tasks like VSR, this classifier significantly underperforms, achieving a relative accuracy of merely 63.5%. It discards a greater number of frames compared to the ground truth result, underscoring its ineffectiveness in selecting frames pivotal for comprehensive task completion.

Regions of Interest (RoI) encoding is inefficient and not robust in video understanding. Other video preprocessing approaches, such as RoI encoding [13], [21], which utilizes a lightweight detector to differentiate between RoI and backgrounds and applies varied encoding configurations, often face limitations in their applicability. Most video understanding tasks lack an explicit definition of RoI. This technique risks overlooking vital details in areas not designated as RoI, potentially compromising the comprehensive understanding required in many video understanding tasks. Furthermore, RoI detection can introduce substantial additional latency, as it lacks the generation of beneficial intermediate results without the need for extra computation, a design previously established in [21]. To quantify this, we evaluated the incremental latency induced by the RoI encoding process across various video understanding tasks, analyzing one-second video segments. We deployed a face detector from the *dlib* library [22], a hand detector from the *mediapipe* library [23], and *deit_base* [24] from the *torch.hub* library as RoI detectors for VSR, VAR and VQA respectively. Our findings indicate that even for tasks such as VSR and VAR, where RoI detection is relatively straightforward, the additional latency incurred was 1.26 seconds and 0.86 seconds, respectively. This process leads to around 3 times more extra latency than before. The more intricate VQA task required a more complex model for RoI extraction, resulting in an untenable RoI encoding duration of 3.65 seconds. Furthermore, while certain video understanding tasks like VSR and VAR benefit from well-defined RoIs and dedicated detectors, others, including VQA and video captioning, do not. The lack of precise RoI definitions in these tasks can lead to significant inaccuracies in encoding, potentially resulting in an approximate 32.5% decrease in accuracy. *This underscores the need for more precise and context-aware preprocessing techniques in video understanding tasks, where traditional RoI encoding falls short.*

Existing streaming designs for video analytics are challenged by sparse feedbacks in video understanding. Approaches focusing on efficient video streaming, such as analytics-aware adaptive bitrate streaming, have traditionally played a significant role in adjusting video bitrates using deep reinforcement learning to align with fluctuating network conditions and content characteristics [15], [16], [25]. These DRL-based methods utilize rewards combining analytics accuracy and communication performance. However, they face limitations in video understanding tasks due to the challenge of sparse feedback. To be specific, feedback for actions taken is delayed, complicating the learning process of the adaptive streaming controller [26], [27]. In video understanding, the effectiveness of analytics cannot be immediately determined

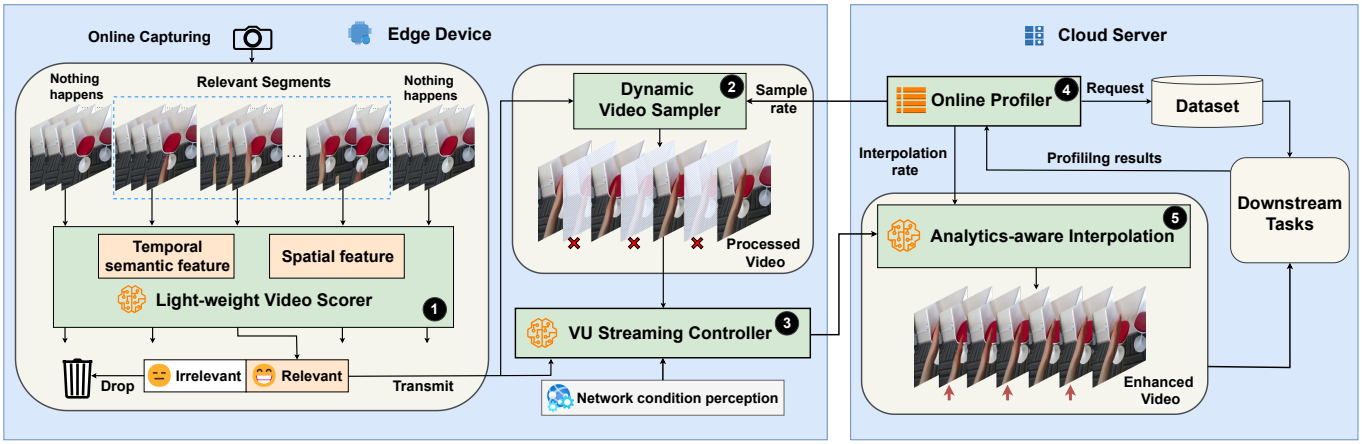


Fig. 3. The edge-cloud collaborative analytics framework for efficient video understanding.

after processing each video segment, as is done in instance-level analytics [28]. We examine the performance of the classical analytics-aware RL approach [16] on adaptive streaming for video understanding applications and find that the performance degradation in analytical accuracy and latency can be more than 40. 3% and 21. 1%, respectively. *Therefore, video streaming controller designs for video understanding tasks must evolve to account for this delayed feedback, necessitating a shift from an immediate evaluation per segment to a more holistic approach that considers entire video segments.*

Other complementary designs miss the optimization room of video understanding. Methods such as efficient backend analytics configuration profiling [29], while optimizing analytics efficiency, often overlook the crucial aspect of communication cost optimization in the video analytics pipeline. Another strategy involves downsampling video resolution at the edge and employing super-resolution techniques in the cloud [14] to recover missed information, effectively reducing communication costs while maintaining high analytics accuracy. However, this method primarily optimizes spatial aspects of video analytics, neglecting potential efficiencies in the temporal domain.

Current video analytics frameworks, while adept at instance-level tasks, reveal significant limitations in addressing the complexities of video understanding tasks. These all call for a new design for video analytics systems to serve video understanding tasks.

III. Vidfra Design and Implementation

A. Overview

The overall architecture is illustrated in Figure. 3. The framework's core philosophy centers on leveraging video content redundancy and cloud computation to achieve a marked reduction in bandwidth usage without sacrificing the accuracy of video inference. This is achieved by employing an adaptive video sampling mechanism on the edge side, with the assistance of extra perceptions on video context and network conditions in real time. Then the downsampled videos will be recovered using analytics-aware interpolation on the server to maintain necessary temporal information for video understanding.

Edge Device Side. In the edge domain, each real-time captured video segment is first preprocessed by a light-weight context-aware video scorer. The context-aware video scorer distinguishes relevant or irrelevant video segments based on their spatial-temporal information. This identifies whether a segment contains task-relevant events. Then the captured video segment is processed by a video sampler. The sampler uniformly samples the video with an adaptive sample rate sent from the profiler on the server side. The sampled video is streamed by an adaptive video streaming controller for video understanding tasks using a sparse-reward PPO algorithm [30]. It dynamically selects the appropriate bitrate for each video segment based on current network conditions.

Cloud Server Side. The cloud server component of Vidfra is responsible for an analytics-aware video frames interpolation process that enhances the processed video to maintain necessary temporal information for video understanding. Recieved videos are enhanced by an analytics-aware video frames interpolation module with an adaptive interpolation rate. The interpolation rate is decided by the profiler as well. The profiler jointly selects the appropriate sampling rate and interpolation rate by profiling. It chooses the configuration that maximizes the analytics accuracy while minimizing the system overhead. This process focuses on maintaining both the video quality and contextual nuances essential for analytics. The detailed design for each component is presented in the following sections.

B. Light-weight Context-aware Video Scoring

To address challenges motivated in Section II, we design and implement a novel lightweight context-aware video scorer module, which enables Vidfra framework's context-aware video sampling on the edge side. The primary objective of this component is to judge the event relevance for each video segment from real-time streaming. In our method, we extract frame-level features for each frame and apply a temporal modeling layer to capture contextual representations jointly that are significant to video understanding.

Given a video segment i of T frames, we first employ ShuffleNetv2 [31], a widely used light-weight backbone network for resource-constraint devices, to extract feature vectors $v_j, j \in \{1, \ldots, T\}$. As the attention mechanism has been proven to be effective in capturing long-range dependencies, it suffers from much higher complexity and resource consumption which is not friendly to edge devices. We adopt Bi-LSTM to extract long-range representations $w_i = \{w_j\}_{j=1}^{T}$ in default. To integrate the contextual representation w_i with the per-frame spatial feature v_j, we use the concatenation of w_i and average of v_j. Namely, the final spatial-temporal representations are computed as $x_i = w_i + \overline{v_j}$. Finally, a classifier block takes the concatenation feature x_i and calculates the probability distribution over binary labels indicating relevant or not as:

$$x_{i'} = \text{ReLU}(W_1^T x_i + b_1), \tag{1}$$

$$p_i = \text{Sigmoid}(W_2^T x_{i'} + b_2), \tag{2}$$

where W_1, W_2 and b_1, b_2 are the parameters of the two FC layers in the classifier block. Additionally, to enhance the suitability for edge devices, input video segments are first downsampled by three times in frame rate to reduce computational demands, a step that has been shown to negligibly impact the classification accuracy. We optimize the video scoring network using offline training and the loss of one input segment can be represented using Binary Cross-Entropy (BCE) loss as:

$$\text{Loss} = -\left[l_i \log(p_i) + (1 - l_i) \log(1 - p_i)\right], \tag{3}$$

where p_i indicates the probability probabilities predicted by the classifier for segment i, l_i represents the ground truth label.

C. Dynamic Video Sampling & Enhancement

Edge-side Video Sampling Following the classification of incoming video segments by the context-aware video scorer, relevant segments undergo further transmission and inference, while irrelevant ones are discarded. These selected segments are initially processed by our dynamic video sampling module, which employs a variable sampling rate s for each segment.

Our implementation involves uniformly sampling $s \times N$ frames from a video segment containing N frames. We also explored alternative discriminative video sampling strategies, such as frame selection. This strategy utilizes an additional neural network to identify the most representative frames within a video segment for more focused inference, potentially reducing the number of frames used. However, we found uniform sampling to be the most efficient and practical approach for several reasons: (1) Frame selection networks add extra inference latency, which can become a bottleneck in edge video processing. (2) This technique requires analysis of the entire video, making it unsuitable for scenarios involving online videos, which lack access to future segments. (3) Selectively chosen frames disrupt the continuity of video sequences, reducing the efficiency of the video encoding process. Although this method selects fewer frames, the need to transmit frames individually results in higher communication costs.

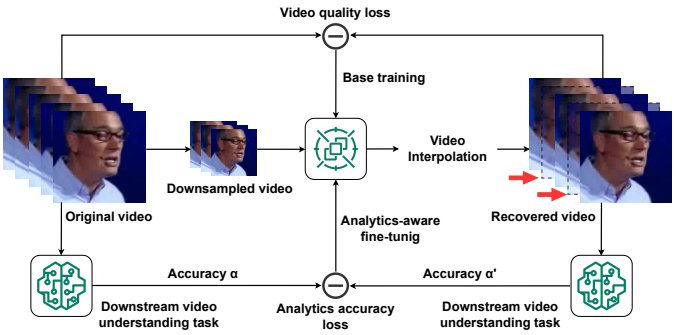


Fig. 4. Illustration of the two-stage analytics-aware training process for video interpolation model. The base interpolation training stage minimizes the video quality loss between predicted interpolated frames and original frames. Subsequently, the analytics-aware tuning stage minimizes the analytical accuracy loss between interpolated videos and original videos.

Server-side Video Enhancement We enhance the sampled videos on the cloud server using our analytics-aware video interpolation module in Vidfra. It reconstructs processed video segments for precise cloud-based analytics. Utilizing a fine-tuned video interpolation model tailored for analytics tasks, this block dynamically interpolates video to preserve contextual information essential for video understanding. Our system utilizes the RIFE model [32] for video interpolation, a state-of-the-art model in Video Frame Interpolation (VFI).

Furthermore, we notice that existing interpolation models, optimized for client experience, often underperform in video understanding due to their focus on visual quality metrics like PSNR and SSIM, which can overlook crucial details regarded as important by vision models. To bridge this gap, we propose a two-stage analytics-aware training process for the video interpolation model. It involves:

- **Base interpolation training:** Originally, RIFE is trained to minimize the quality loss (structural similarity index, or SSIM) between the original video frames and the resulting interpolated video frames.
- **Analytics-aware fine-tuning:** Next, we further train the interpolation model to improve the accuracy of specific vision tasks. We calculate the difference in inference accuracy between running the vision model on the original videos and running it on the interpolated videos. This difference is then used as the loss function which the new interpolation model is trained to minimize.

The analytics-aware video interpolation block is adeptly integrated within the system, operating with high efficiency. Importantly, as revealed in our previous Motivation section and confirmed by our evaluations, sometimes it is unnecessary to reconstruct all omitted frames, opening avenues for optimizing the interpolation rate. The optimal rate for each received video is determined through a joint profiling approach along with the sampling rate which will be introduced in the next section. The interpolation process has been proven through evaluation, to run within our system at 30 fps approximately, thus aligning with the demands of real-time video analytics.

Joint Configuration Tuning via Online Profiling During

the configuration profiling stage, our objective is to identify an optimal pair of settings: the video sampling rate and the video interpolation rate. These rates are tuned together to achieve three primary goals: (1) minimize video transmission costs by using the lowest feasible sampling rate; (2) reduce server-side interpolation overhead by applying the minimal necessary interpolation rate; and (3) maintain or enhance analytical accuracy on post-processed videos.

In detail, the profiler evaluates several minutes of video for each candidate configuration pair (sample rate and interpolation rate). It processes these videos through sampling and enhancement procedures, then calculates the analytical accuracy on downstream tasks. Based on these outcomes, the profiler selects the most suitable configuration pair, which is then applied to the edge-side video sampling and cloud-side video interpolation modules.

This integrated approach, combining edge-side video sampling with server-side video enhancement, effectively alleviates the communication bottleneck between edge and cloud. We observed that simple adjustments, such as sampling or modifying the frame rate at the edge, can reduce communication costs while preserving acceptable cloud-end analytical accuracy. However, these adjustments alone are insufficient and not universally applicable across different tasks. The synergistic operation of edge-side sampling and server-side enhancement not only allows for more significant reductions in transmission costs but also increases the tolerance for sparser video frame sampling. This is because server-side enhancement can reconstruct some dropped frames, thereby accommodating a higher degree of sparsity without compromising analytical accuracy. In contrast, merely adjusting the sampling or frame rate at the edge without server-side enhancements lacks this tolerance, and aggressive sampling could directly impair analytical performance.

D. Adaptive Streaming Controller for Video Understanding

We propose an adaptive streaming controller to handle aforementioned unique challenges in streaming for video understanding tasks. This controller dynamically adjusts video streaming configurations according to real-time network conditions for each video segment, such as bitrate. However, naively integrating RL suffers from sparse feedback during the interaction between the agent and the environment, which leads to misleading short-term rewards and poor learning performance (Recall 40% and 21.1% accuracy and latency degradation as we stated in Section II).

Therefore, we formulate the network-aware adaptive configuration selection for each video segment c as a Markov Decision Process, and design state, action, reward, and training algorithm to cope with this challenge as follows.

- **State space s_c:** timestamp t, buffer occupancy o_t, real-time predicted bandwidth b_t by an LSTM-based bandwidth predictor, past action a_{c-1} and past segment size s_{c-1}.
- **Action space a_c:** the agent selects an appropriate encoding bitrate br_c from a set of options.

Delayed reward modeling. We utilize credit assignment technique for delayed reward modeling. The agent aims at maximizing video analytics accuracy while minimizing communication cost. Especially, the complete reward can only be calculated after all N segments of a complete video segment are streamed. The reward consists of two parts: an intermediate reward r_i and a delayed accuracy reward r_d. The intermediate reward depends on the transmission latency l_i and the bandwidth consumption β_i of each segment. The delayed accuracy reward depends on the video segment's analytics accuracy α. After a complete video is streamed and analyzed, the delayed accuracy reward will be assigned to each segment in this video. The contribution to the accuracy reward of each segment is determined by the relevance score s_i, which is calculated by the video scorer in Eq. (2). Then we calculate the delayed accuracy reward of each segment as $\alpha_i = \alpha \times \frac{s_i}{\sum_N s_i}$. The overall reward function can be present as $\omega_1 \cdot \alpha_i - (\omega_2 \cdot l_i + \omega_3 \cdot \beta_i)$.

Network architecture. The agent takes actions based on the policy $\pi_{\theta_a}{:}p(s_c, a_c) \longrightarrow [0, 1]$, representing the probability of taking an action a_c at state s_c. θ_a are the parameters of the actor network. To be specific, the actor network consists of one FC layer and an activation layer using the softmax function. The critic network consists of two FC layers. As for the input state, past action and past segment size are processed by a 1DCNN layer and then concatenated with other state components to form the embedding vector, which is used as the input for the actor and critic networks. We use the PPO method [30] as our training algorithm. Given θ_a, the gradient of the accumulated discounted reward can be calculated as:

$$\nabla_{\theta_a} E_{\pi_{\theta_a}} \left[\sum_{c=0}^{n} \gamma^c r^c \right] = E_{\pi_{\theta_a}} \left[\nabla_{\theta_a} \log \pi_{\theta_a}(s,a) A_{s,a}^{\pi_{\theta_a}} \right], \quad (4)$$

$$A_{s,a}^{\pi_{\theta_a}} = r_c + \gamma V_{\theta_v}^{\pi_{\theta_a}}(s_{c+1}) - V_{\theta_v}^{\pi_{\theta_a}}(s_c), \quad (5)$$

where γ is the discounting factor on future rewards, $A^{\pi_{\theta_a}}(s,a)$ is the advantage function over action a and $V_{\theta_v}^{\pi_{\theta_a}}(s_c)$ is the output of the critic network. The actor and critic network parameters can be updated as:

$$\theta_a = \theta_a + \eta_a \sum_c \nabla_{\theta_a} \log \pi_{\theta_a}(s,a) A_{s,a}^{\pi_{\theta_a}} + \delta \nabla_{\theta_\alpha} H_{s_c}^{\pi_{\theta_\alpha}}, \quad (6)$$

$$\theta_v = \theta_v - \eta_v \sum_c \nabla_{\theta_a} \left[A_{s_c,a_c}^{\pi_{\theta_a}} \right]^2, \quad (7)$$

where η_a, η_v is the learning rate for the actor and critic network respectively. $H(\cdot)$ is the entropy of policy to discourage converging to a sub-optimal policy.

To address the challenge of sparse feedback in the video streaming scenario for video understanding, we propose our delayed-reward training algorithm, as described in Algorithm 1. This strategy incorporates both Intermediate Rewards (IR) and Delayed Accuracy Rewards (DAR). IRs are granted based on real-time indicators such as maintaining optimal buffer health or bitrate ranges, providing immediate feedback to the system. The DARs are evaluated after N segments, which is the

Algorithm 1: PPO Algorithm for Policy Training

Result: Optimized policy parameters θ_{actor} and θ_{critic}
Input: Learning rate α, discount factor γ, number of episodes M, steps per episode C_{max}, environment $\mathcal{E}$, actor network architecture $\mathcal{A}_{\text{actor}}$, critic network architecture $\mathcal{A}_{\text{critic}}$, clipping parameter ϵ, number of epochs K, mini-batch size N_{batch};
Initialize: Actor network parameters θ_{actor}, Critic network parameters θ_{critic}, Replay buffer $\mathcal{B}$;
for $episode = 1$ **to** M **do**
  Initialize sequence buffer $\mathcal{S}$;
  Receive initial state s_1 from environment $\mathcal{E}$;
  for $t = 1$ **to** C_{max} **do**
    Sample action a_t from policy $\pi(\cdot|s_t; \theta_{\text{actor}})$;
    Execute action a_t in environment $\mathcal{E}$;
    Observe the reward r_i and new state s_{t+1};
    Store transition in sequence buffer $\mathcal{S}$;
    if *t mod N == 0 or end of episode* **then**
      Compute reward r_d with delayed feedback for the sequence;
      Compute accuracy contribution s_i for each segment in the sequence;
      For each (s, a, r_i, s') in $\mathcal{S}$, adjust r_i based on each segment's accuracy contribution and store (s, a, r_i, s') in $\mathcal{B}$;
      Empty sequence buffer $\mathcal{S}$;
    end
  end
  Sample mini-batch of size N_{batch} from $\mathcal{B}$;
  for $k = 1$ **to** K **do**
    Compute advantage estimates $\hat{A}_t$ using critic network and clipped surrogate objective;
    Update θ_{actor}
    Update θ_{critic}
  end
end

point at which all segments in a complete video segment finish transmission. This delayed feedback mechanism evaluates valid rewards for sparse feedback of accuracy in the video streaming scenario for video understanding tasks.

During training episodes, the cumulative reward for each episode is calculated as the sum of IRs and DAR. This cumulative reward then guides the updates to the actor and critic networks in the PPO algorithm. The actor network adjusts its policy to maximize these returns, while the critic network refines its predictions of expected rewards. After being processed by the adaptive video streaming controller, the processed video segments are transmitted along with the scene change information to the cloud server for further processing and analytics.

IV. Evaluation

A. Experiment Setup

In our evaluation, we considered three typical classes of video understanding tasks which have different sensitivity to temporal information: (1) **Video Action Recognition** involves classifying actions within videos. (2) **Video Speech Recognition** involves decoding text from the movement of speakers' lips from silent videos, namely lip reading. (3) **Video Activity Detection** involves detecting and classifying human activities within videos.

Datasets: (1) **Video datasets**: We choose NVGesture [33], LRS3-TED [34] and ActivityNet v1.3 [35] as our evaluation datasets of video understanding tasks. To be specific, NVGesture is for the open domain video action recognition, with 1050 video clips, 25 different actions along with action boundaries; NVGesture's video frame resolution is (320,240), frame rate is 30 fps, mean bitrate is 5.80 Mbps; LRS3-TED is a multi-modal dataset for lip reading which includes over 400 hours of TED and TEDx videos, along with speaking boundaries; LRS3-TED's video frame resolution is (224,224), frame rate is 25 fps, mean bitrate is 2.70 Mbps; ActivityNet v1.3 is the largest temporal activity detection dataset, which contains 200 different types of activities annotated with temporal boundaries; ActivityNet's videos have various resolution and frame rate, the mean bitrate of selected videos is 1.92 Mbps; (2) **Network datasets**: We chooise 4G LTE dataset [36] and 5G dataset [37] for video streaming simulation under different network traces. 4G LTE and 5G network traces are collected across video streaming application patterns, and several mobility patterns such as indoor, car, and train, with an average duration of over ten minutes per trace.

Benchmarks: We compare Vidfra with the following benchmarks.

- **Baseline:** It represents the streaming performance and analytics result of the original video without any additional processing strategy.
- **CASVA_M [28]:** This approach adjust frame rate (FR) from the temporal perspective, resolution (RS) from the spatial perspective, and quantization parameter (QP) that controls the image quality in real-time on the edge. We adapt the streaming controller to the same video understanding version as ours.
- **SMS [38]:** It is a state-of-the-art approach that introduces an frame sampling approach for video recognition. It develops a neural network to filter redundant video frames and selectively choose the most representative video frames that may achieve the best results for inference.
- **Reducto [12]:** This approach deploys a lightweight frame filter on edge devices that utilizes low-level vision features for fast frame selection. It makes a binary decision on receiving each video frame and it adjusts the filter threshold by online retraining.
- **CloudSeg [14]:** This approach involves downsampling video resolution at the edge then reconstructing using super-resolution at the cloud server. It also deploys a

two-stage frame filter based on pixel difference between adjacent frames.

Parameter settings: For the light-weight video scorer, the hidden size of Bi-LSTM is 1024; the number of LSTM layers is 2; the hidden size of FC layer is 512; For the video sampler, the sample rate candidates are [$\frac{1}{2}, \frac{1}{3}, \frac{1}{4}, \frac{1}{6}, \frac{1}{8}$]; For the adaptive video streaming pipeline, we consider actions and segment sizes of past 8 segments; the number of filters in 1DCNN is 128, its stride and kernel size is 1 and 4, respectively; the hidden size of FC layers is 128; the learning rate of both action policy network and critic policy network is 0.0001; the discount factor is 0.9; the hidden size of bandwidth prediction LSTM is 64; the number of the LSTM layers is 2; the bitrate candidates are the bitrate for [240P, 360P, 480P, 720P, 1080P]; For the analytics-aware interpolation block on the cloud server, the video interpolation rate candidates are [2,3,4,6,8]; For the backend video analytics, the three selected video understanding tasks are evaluated by [18], [39], [40] respectively; The edge side system uses a Jetson TX2 platform with a 256-core NVIDIA Pascal GPU; The backend cloud server has four 3090Ti GPUs and a 24-core Intel Xeon Gold 5119T CPU.

Metrics: We use the following metrics to examine `VidFra`'s performance in analytic accuracy, and the efficiency of the analytic system.

- **Mean bandwidth usage.** We calculate the mean bandwidth usage for all test videos. The usage ratio calculates the result of the transmitted video size divided by the original video size. On the contrary, the mean bandwidth saving ratio is derived by subtracting the mean bandwidth usage ratio from 1.
- **Mean latency.** We calculate the mean end-to-end latency for all test videos. This includes the pre-processing (video scoring, video sampling, adaptive controller inference, video encoding) latency, video streaming latency and post-processing (analytics-aware interpolation) latency.
- **Mean accuracy.** We calculate the mean inference accuracy at the cloud end for all test videos. The accuracy of VAR is evaluated using the proportion of correctly identified actions. The accuracy of VSR can be measured using one minus word error rate (WER). The accuracy of VAD is evaluated using the mAP50 of detected activities.

B. Experiment Results

Bandwidth consumption vs. accuracy. Fig. 5(a), 5(b), 5(c) illustrates the tradeoff between accuracy and bandwidth for the six compared approaches. The results demonstrate that Vidfra surpasses other methods in achieving the most favorable accuracy-bandwidth trade-off. For VAR task, Vidfra significantly reduces bandwidth consumption by as much as 57.7% at an equivalent accuracy level. For VSR task, Vidfra significantly reduces bandwidth consumption by as much as 52.2% at an equivalent accuracy level. Furthermore, we notice that Vidfra is much more flexible than other benchmarks. Other benchmarks can only save no more than 24.6% bandwidth while reducing the accuracy requirement by over 31.2%. Vidfra can save as much as 35.2% and 49.8% bandwidth usage while sacrificing less than 15% accuracy for VAR and VSR tasks, respectively. These results underscore Vidfra's superior efficiency in conserving bandwidth in scenarios constrained by network resources.

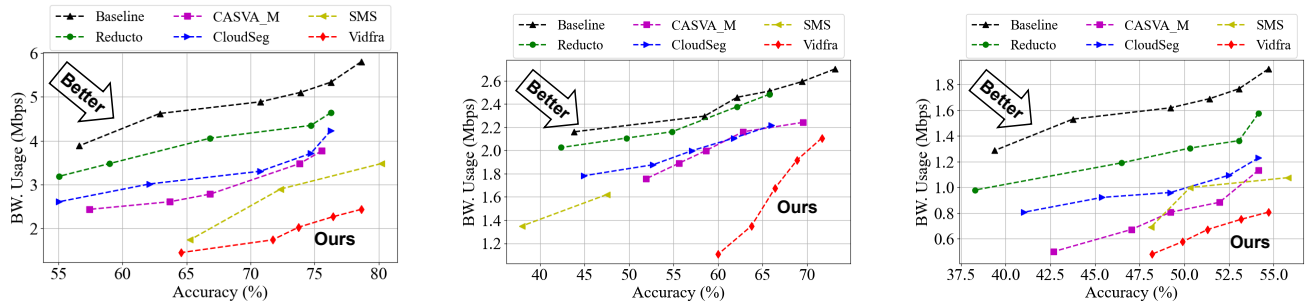
End-to-end latency vs. accuracy. Fig. 6(a), 6(b), 6(c) shows the trade-off between accuracy and end-to-end latency for the six compared approaches. The trade-offs are achieved by adjusting configuration parameters such as sampling rate, interpolation rate. It can be observed that Vidfra achieves the best accuracy-latency trade-off compared to other approaches. For VAR task, Vidfra significantly outperforms other approaches by reducing the end-to-end latency by up to 52.8%. For VSR task, Vidfra significantly outperforms other approaches by reducing the end-to-end latency by up to 28.1% while maintaining the same accuracy levels. Vidfra significantly reduces the end-to-end latency compared to state-of-the-art frame filter, frame selection and adaptive frame rate techniques in terms of accuracy. These findings underscores Vidfra's exceptional capability in minimizing end-to-end latency while achieving high accuracy.

System sensitivity & overhead: To examine the sensitivity and overhead of our system, we evaluate the bandwidth saving ratio and end-to-end response time for various segment sizes from short bursts to longer sequences as 1, 5, 10, 30, and 60 seconds. The trends are illustrated in Figure. 7.

In our study as shown in Figure. 7(a), we noted that for smaller segments (1 second), our system's bandwidth savings are around 51.9%, a figure that remains stable across various segment lengths, indicating a robust filtering mechanism for various segment durations. As segment size increases, we observed a rise in savings, with 60-second segments achieving up to 74.3% savings. This suggests our algorithm's ability to apply more effective compression in longer segments, balancing data reduction with content integrity. Additionally, extended segments improve interpolation by providing more contextual frames, further increasing bandwidth savings.

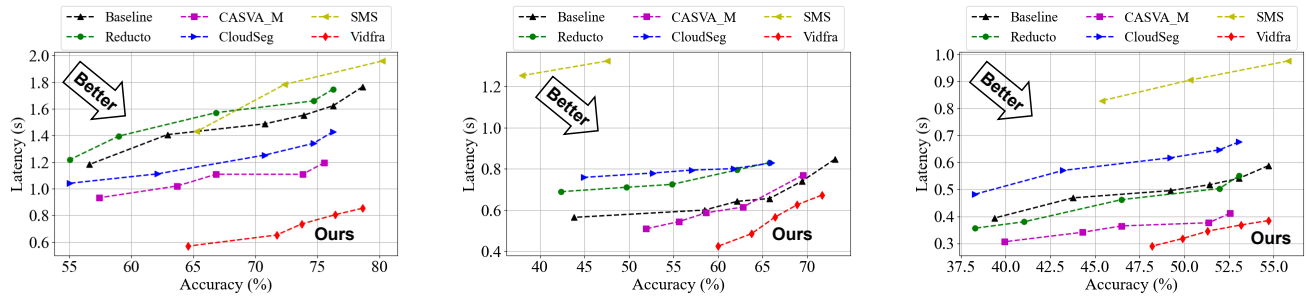
We evaluated the system overhead of each component. To be specific, the throughput of video scoring is 308.3fps, which costs only 0.096 seconds for one second video segment; the policy decision and video encoding process is 87.5fps and the video interpolation process is 29.6fps. We further present the end-to-end response time in Figure. 7(b). The response time increases with segment size, but not proportionally, indicating that longer segments transmit more efficiently. Specifically, a 1-second segment has a response time of just over 0.55 seconds, increasing to about 11.56 seconds for 60-second segments. This supports our observation of improved bandwidth savings in longer segments. Transmission latency is the main contributor to end-to-end latency, at 69.5%. Edge-side preprocessing latency, involving tasks such as event detection and encoding, remains below 16.4% across segment sizes, reflecting our system's consistent performance over various segment lengths.

Network-aware adaption. We also test the latency of one-second segment transmission varies significantly with network technology. Under 4G LTE conditions, our approach shows an approximate latency of 0.32 seconds, which is 44.9% lower



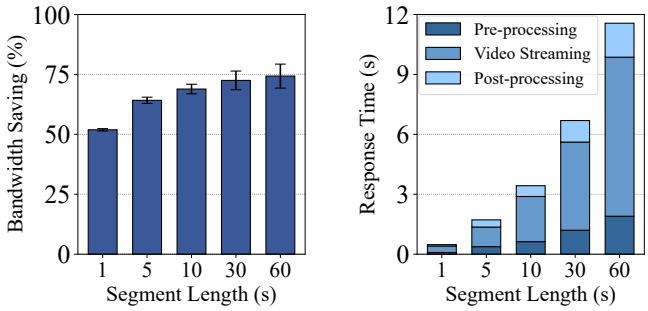
(a) Video Action Recognition (b) Video Speech Recognition (c) Video Activity Detection

Fig. 5. The overall bandwidth usage versus accuracy performance comparison on selected video understanding tasks.



(a) Video Action Recognition (b) Video Speech Recognition (c) Video Activity Detection

Fig. 6. The overall end-to-end latency versus accuracy performance comparison on selected video understanding tasks.



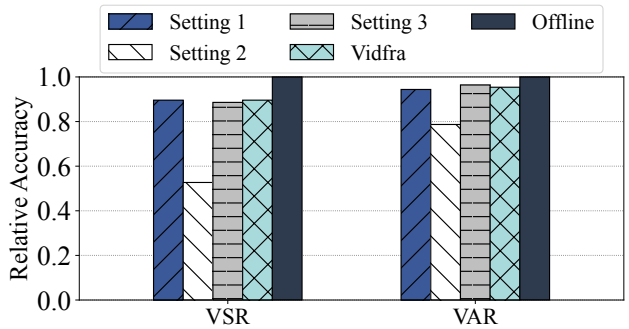
(a) (b)

Fig. 7. Distribution of response time and bandwidth saving ratio on different video segment lengths. Response time consists of pre-processing (video scoring, video sampling, adaptive controller inference, video encoding) latency, video streaming latency and post-processing (analytics-aware interpolation) latency. Results are simulated under our 4G LTE traces.

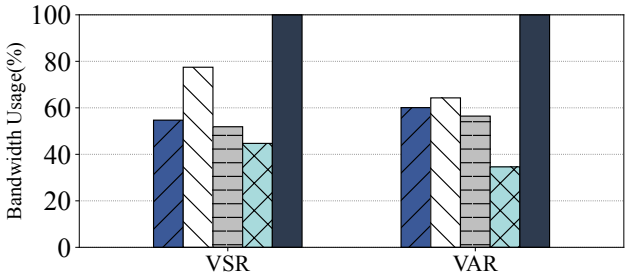
than other benchmarks. When operating over 5G networks, our method's latency advantage is also pronounced, maintaining a similar latency profile to 4G LTE, despite the increased bandwidth capabilities, suggesting our algorithm efficiently adapts to diverse network conditions. The comparative latency under both network conditions indicates that our adaptive bitrate controller, informed by real-time network analytics such as bandwidth and network latency, and application-layer metrics like inference accuracy and computation time, effectively calibrates the video streaming parameters.

Ablation study. We propose three decomposed settings of VidFra to examine the contribution of each block in our system. *Setting 1* deletes the video scoring module, which processes all generated video segments. *Setting 2* deletes the video interpolation module on the server side. It only conducts adaptive video sampling on the edge side. *Setting 3* deletes the joint profiler module, which conducts fixed video sampling on the edge side and tries to interpolate all dropped frames on the server side. The performance of each configuration is detailed in Fig. 8.

As for setting 1, it keeps similar analytical accuracy compared to our result and offline result because of transmitting all generated video segments. However, the bandwidth usage of setting 1 is 21. 4% higher on average than ours. It reveals the effectiveness of the video scoring module in reducing irrelevant video segments. As for setting 2, it leads to a 30.5% accuracy drop compared to our result. It also leads to a 31.9% higher bandwidth usage compared to our result. This suggests that naive video downsampling significantly damages temporal information, leading to poor accuracy. An analytics-aware video interpolation block can effectively address this loss and save much more bandwidth usage. As for setting 3, it achieves similar analytical accuracy compared to our result. However, the bandwidth usage is 15.6% higher than our result on average. This indicates the effectiveness of the joint profiler

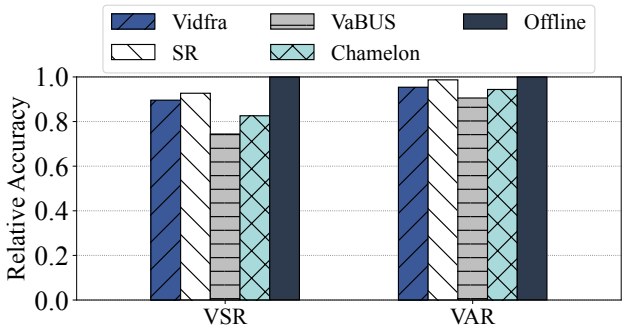


(a) Mean accuracy compared to the offline optimal accuracy of original videos.

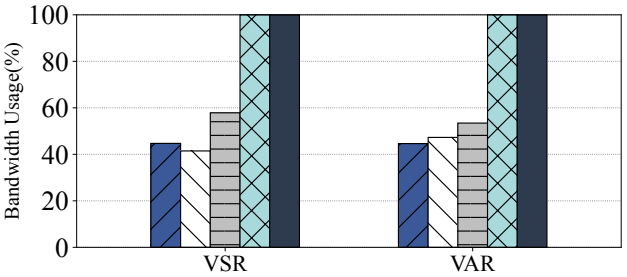


(b) Mean bandwidth usage compared to the bandwidth usage of original videos.

Fig. 8. The ablation study over accuracy and bandwidth usage performance of different system settings on selected video understanding tasks.



(a) Mean accuracy compared to the offline optimal accuracy of original videos.



(b) Mean bandwidth usage compared to the bandwidth usage of original videos.

Fig. 9. The overall accuracy and bandwidth usage performance of different comparison methods on selected video understanding tasks.

in dynamically selecting appropriate configurations for video sampling and video interpolation modules.

Comparison with complementary approaches: As illustrated in 9, we compared Reducto with three methodologies that enhance the efficiency of real-time video analytics pipelines: Super Resolution (SR) [14], Chameleon [29], and VaBUS [41]. Each system targets a different component of the analytics pipeline, yet conceptually, all are complementary to Vidfra.

SR [14]. SR involves downsampling video resolution at the edge before transmission and applying analytics-aware super-resolution techniques at the cloud server to reconstruct the original video frames. We opted not to use its two-stage frame filter, intended for keyframe selection, as it significantly impairs video understanding tasks. We employed a downsampling rate of 2x in our experiments. Although SR demonstrates competitive accuracy and bandwidth efficiency comparable to Vidfra, the super-resolution process at the backend introduces significant latency, approximately $2.3\times$ that of our method.

Chameleon [29]. Chameleon aims to reduce backend computational costs by profiling various pipeline configurations (e.g., video resolution, frame sampling rate) and selecting those that optimize resource use while fulfilling specified accuracy criteria. In our adaptations, we dynamically adjust the video frame rate and resolution based on video profiling over several minutes. Chameleon achieves approximately 5.6% lower accuracy than our method due to its sole reliance on downsampling the frame rate, which disrupts temporal dependencies. Moreover, by only modifying backend analytics configurations, Chameleon fails to reduce bandwidth consumption during network transmission.

VaBUS [41]. VaBUS dynamically manages the video's background on the edge with minimal system impact and transmits only the regions of interest (RoIs) with high confidence to the cloud, using adaptive weighting and encoding. In our tests, it isolated hand regions for VAR tasks and mouth areas for VSR tasks using detection models. VaBUS shows 8.2% lower accuracy and 18.2% increased bandwidth usage compared to our method. Additionally, its approach is less applicable to advanced video understanding tasks, such as video captioning and video question answering, which lack well-defined RoIs. The RoIs identification process also contributes to a 67.3% increase in end-to-end latency.

V. Related Work

Edge-Cloud Collaborative Video Analytics System. The rise of edge devices in distributed locations has made the development of efficient edge-cloud collaborative analytics frameworks a key focus for real-time video analytics. Balancing the transmission of high-fidelity video data to the cloud for analysis without excessive bandwidth use, while ensuring high analytical accuracy, presents a significant challenge.

Recent advancements have seen several efficient video analytics systems. For example, systems such as Reducto [12], FilterForward [19], and Glimpse [11] employ edge-based frame filtering techniques to preprocess keyframes. Reducto utilizes basic computer vision features for frame filtering, while FilterForward and Glimpse use lightweight neural networks as binary classifiers. In terms of scalability, CloudSeg [14] addresses spatial information in frames, combining frame

filtering with super-resolution techniques to reduce bandwidth. Similarly, DDS [13] and DRE [21] focus on minimizing spatial redundancy through unique RoI related streaming strategies. Others like AWStream [15], and Chameleon [29] emphasize adaptive video streaming policies, data rate adjustments, and computational resource management for large-scale streaming analytics. However, as outlined in Section II, the focus of existing systems on instance-level analytics limits their effectiveness for video understanding tasks. Vidfra, our system, uniquely fills this gap as the first designed for video understanding tasks, addressing temporal redundancy and enhancing streaming efficiency under diverse network conditions.

Video Understanding. Distinct from instance-level tasks such as object detection and semantic segmentation, video understanding, or sequence-level video analytics, necessitates interpreting a sequence of frames. Specialized DNN architectures and analytical methods are proposed to effectively process frame sequences. For instance, 3DCNN [5] and TSN [6] add an additional temporal dimension to the spatial input channels of traditional CNNs, enabling the process of both spatial and temporal data. Beyond these, temporal analytical networks, such as LSTM [42] and attention mechanisms [43], are also employed to interpret sequences of frame features effectively.

Recent research has also focused on improving the efficiency of video understanding. TSM [44] and Eco [45] proposed unique network structures for video understanding that balances analytical performance and computational economy. VideoIQ [46] and ResQ [47] proposed novel quantization approaches that effectively reduce the computational burden by optimizing the precision of network parameters. Despite these advances in the machine learning field, providing video understanding tasks with efficient system support has barely been studied. Our work reveals the limitations of existing video analytics systems to serve video understanding tasks and proposes a novel system to fill this gap.

Video Frames Interpolation. Our system partially builds upon the advancements in video frame interpolation (VFI) techniques [48], which reconstruct intermediate scenes from adjacent frames by inferring optical flow [32], [49]. Recent improvements in DNN-based VFI also try to meet real-time requirements [32] and have been applied in video streaming applications to enhance user's quality of experience [50]. Our work stands out in two main ways. First, we explore the potential of applying VFI to downsampled videos to run video understanding tasks from a system perspective, effectively balancing bandwidth saving in encoding with accuracy in analytics. Second, unlike the current VFI models that focus on visual quality for human viewers, we propose an analytics-aware training approach for VFI models that optimizes both post-VFI visual quality and video analytics accuracy.

VI. Conclusion

In this paper, we present Vidfra, the first edge-to-cloud framework revolutionizing real-time video understanding. By efficiently balancing bandwidth consumption and inference accuracy, Vidfra addresses the challenges of temporal information loss and network resource limitations prevalent in traditional video analytics pipelines. Its unique approach of selective frame processing, coupled with analytics-aware video interpolation, demonstrates a significant improvement in both bandwidth savings and accuracy across various video understanding tasks. The experimental results validate Vidfra's superiority over existing methods, highlighting its potential as a robust, adaptable solution for efficient video understanding, particularly in bandwidth-constrained environments. Vidfra's novel design and comprehensive performance establish it as a pioneering system in the realm of video understanding, calling for more research efforts in this field from a system perspective.

References

[1] S. R. E. Datondji, Y. Dupuis, P. Subirats, and P. Vasseur, "A survey of vision-based traffic monitoring of road intersections," *IEEE transactions on intelligent transportation systems*, vol. 17, no. 10, pp. 2681–2698, 2016.

[2] S. Vishwakarma and A. Agrawal, "A survey on activity recognition and behavior understanding in video surveillance," *The Visual Computer*, vol. 29, pp. 983–1009, 2013.

[3] J. Liu, Z. Huang, H. Cai, H. T. Shen, C. W. Ngo, and W. Wang, "Near-duplicate video retrieval: Current research and future trends," *ACM Computing Surveys (CSUR)*, vol. 45, no. 4, pp. 1–23, 2013.

[4] D.-A. Huang, V. Ramanathan, D. Mahajan, L. Torresani, M. Paluri, L. Fei-Fei, and J. C. Niebles, "What makes a video a video: Analyzing temporal information in video understanding models and datasets," in *Proc. IEEE CVPR*, 2018, pp. 7366–7375.

[5] S. Ji, W. Xu, M. Yang, and K. Yu, "3d convolutional neural networks for human action recognition," *IEEE TPAMI*, vol. 35, no. 1, pp. 221–231, 2012.

[6] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool, "Temporal segment networks: Towards good practices for deep action recognition," in *Proc. Springer ECCV*, 2016, pp. 20–36.

[7] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić, and C. Schmid, "Vivit: A video vision transformer," in *Proc. IEEE/CVF ICCV*, 2021, pp. 6836–6846.

[8] Z. Liu, J. Ning, Y. Cao, Y. Wei, Z. Zhang, S. Lin, and H. Hu, "Video swin transformer," in *Proc. IEEE/CVF CVPR*, 2022, pp. 3202–3211.

[9] D. Neimark, O. Bar, M. Zohar, and D. Asselmann, "Video transformer network," in *Proc. IEEE/CVF ICCV*, 2021, pp. 3163–3172.

[10] R. Girdhar, J. Carreira, C. Doersch, and A. Zisserman, "Video action transformer network," in *Proc. IEEE/CVF CVPR*, 2019, pp. 244–253.

[11] T. Y.-H. Chen, L. Ravindranath, S. Deng, P. Bahl, and H. Balakrishnan, "Glimpse: Continuous, real-time object recognition on mobile devices," in *Proc. ACM Sensys*, 2015, pp. 155–168.

[12] Y. Li, A. Padmanabhan, P. Zhao, Y. Wang, G. H. Xu, and R. Netravali, "Reducto: On-camera filtering for resource-efficient real-time video analytics," in *Proc. ACM SIGCOMM*, 2020, pp. 359–376.

[13] K. Du, A. Pervaiz, X. Yuan, A. Chowdhery, Q. Zhang, H. Hoffmann, and J. Jiang, "Server-driven video streaming for deep learning inference," in *Proc. ACM SIGCOMM*, 2020, pp. 557–570.

[14] Y. Wang, W. Wang, J. Zhang, J. Jiang, and K. Chen, "Bridging the {Edge-Cloud} barrier for real-time advanced vision analytics," in *Proc. USENIX HotCloud*, 2019.

[15] B. Zhang, X. Jin, S. Ratnasamy, J. Wawrzynek, and E. A. Lee, "Awstream: Adaptive wide-area streaming analytics," in *Proc. ACM SIGCOMM*, 2018, pp. 236–252.

[16] H. Mao, R. Netravali, and M. Alizadeh, "Neural adaptive video streaming with pensieve," in *Proc. ACM SIGCOMM*, 2017, pp. 197–210.

[17] Y. Zhang, C. Cao, J. Cheng, and H. Lu, "Egogesture: A new dataset and benchmark for egocentric hand gesture recognition," *IEEE Transactions on Multimedia*, vol. 20, no. 5, pp. 1038–1050, 2018.

[18] Y. Jang, Y. Song, C. D. Kim, Y. Yu, Y. Kim, and G. Kim, "Video Question Answering with Spatio-Temporal Reasoning," *IJCV*, 2019.

[19] C. Canel, T. Kim, G. Zhou, C. Li, H. Lim, D. G. Andersen, M. Kaminsky, and S. Dulloor, "Scaling video analytics on constrained edge nodes," *MLsys*, vol. 1, pp. 406–417, 2019.

[20] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," in *Proc. IEEE CVPR*, 2017, pp. 7263–7271.

[21] L. Liu, H. Li, and M. Gruteser, "Edge assisted real-time object detection for mobile augmented reality," in *Proc. ACM MobiCom*, 2019, pp. 1–16.

[22] D. E. King, "Dlib-ml: A machine learning toolkit," *JMLR*, vol. 10, pp. 1755–1758, 2009.

[23] C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C.-L. Chang, M. G. Yong, J. Lee *et al.*, "Mediapipe: A framework for building perception pipelines," *arXiv preprint arXiv:1906.08172*, 2019.

[24] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," in *Proc. PMLR ICML*, 2021, pp. 10 347–10 357.

[25] H. Yeo, Y. Jung, J. Kim, J. Shin, and D. Han, "Neural adaptive content-aware internet video delivery," in *Proc. USENIX OSDI*, 2018, pp. 645–661.

[26] M. Riedmiller, R. Hafner, T. Lampe, M. Neunert, J. Degrave, T. Wiele, V. Mnih, N. Heess, and J. T. Springenberg, "Learning by playing solving sparse reward tasks from scratch," in *Proc. PMLR ICML*, 2018, pp. 4344–4353.

[27] B. Han, Z. Ren, Z. Wu, Y. Zhou, and J. Peng, "Off-policy reinforcement learning with delayed rewards," in *Proc. PMLR ICML*, 2022, pp. 8280–8303.

[28] M. Zhang, F. Wang, and J. Liu, "Casva: Configuration-adaptive streaming for live video analytics," in *Proc. IEEE INFOCOM*, 2022, pp. 2168–2177.

[29] J. Jiang, G. Ananthanarayanan, P. Bodik, S. Sen, and I. Stoica, "Chameleon: scalable adaptation of video analytics," in *Proc. ACM SIGCOMM*, 2018, pp. 253–266.

[30] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[31] X. Zhang, X. Zhou, M. Lin, and J. Sun, "Shufflenet: An extremely efficient convolutional neural network for mobile devices," in *Proc. IEEE CVPR*, 2018, pp. 6848–6856.

[32] Z. Huang, T. Zhang, W. Heng, B. Shi, and S. Zhou, "Real-time intermediate flow estimation for video frame interpolation," in *Proc. Springer ECCV*, 2022, pp. 624–642.

[33] P. Molchanov, X. Yang, S. Gupta, K. Kim, S. Tyree, and J. Kautz, "Online detection and classification of dynamic hand gestures with recurrent 3d convolutional neural network," in *Proc. IEEE CVPR*, 2016, pp. 4207–4215.

[34] T. Afouras, J. S. Chung, and A. Zisserman, "Lrs3-ted: a large-scale dataset for visual speech recognition," *arXiv preprint arXiv:1809.00496*, 2018.

[35] F. Caba Heilbron, V. Escorcia, B. Ghanem, and J. Carlos Niebles, "Activitynet: A large-scale video benchmark for human activity understanding," in *Proc. IEEE/CVF CVPR*, 2015, pp. 961–970.

[36] D. Raca, J. J. Quinlan, A. H. Zahran, and C. J. Sreenan, "Beyond throughput: A 4g lte dataset with channel and context metrics," in *Proc. ACM MMSys*, 2018, pp. 460–465.

[37] D. Raca, D. Leahy, C. J. Sreenan, and J. J. Quinlan, "Beyond throughput, the next generation: a 5g dataset with channel and context metrics," in *Proc. ACM MMSys*, 2020, pp. 303–308.

[38] M. Zhao, Y. Yu, X. Wang, L. Yang, and D. Niu, "Search-map-search: A frame selection paradigm for action recognition," in *Proc. IEEE/CVF ICCV*, 2023, pp. 10 627–10 636.

[39] B. Shi, W.-N. Hsu, K. Lakhotia, and A. Mohamed, "Learning audio-visual speech representation by masked multimodal cluster prediction," in *Proc. ICLR*, 2021.

[40] D. Shi, Y. Zhong, Q. Cao, L. Ma, J. Li, and D. Tao, "Tridet: Temporal action detection with relative boundary modeling," in *Proc. IEEE/CVF CVPR*, 2023, pp. 18 857–18 866.

[41] H. Wang, Q. Li, H. Sun, Z. Chen, Y. Hao, J. Peng, Z. Yuan, J. Fu, and Y. Jiang, "Vabus: Edge-cloud real-time video analytics via background understanding and subtraction," *IEEE JSAC*, vol. 41, no. 1, pp. 90–106, 2022.

[42] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[43] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[44] J. Lin, C. Gan, and S. Han, "Tsm: Temporal shift module for efficient video understanding," in *Proc. IEEE/CVF ICCV*, 2019, pp. 7083–7093.

[45] M. Zolfaghari, K. Singh, and T. Brox, "Eco: Efficient convolutional network for online video understanding," in *Proc. Springer ECCV*, 2018, pp. 695–712.

[46] X. Sun, R. Panda, C.-F. R. Chen, A. Oliva, R. Feris, and K. Saenko, "Dynamic network quantization for efficient video inference," in *Proc. IEEE/CVF ICCV*, 2021, pp. 7375–7385.

[47] D. Abati, H. Ben Yahia, M. Nagel, and A. Habibian, "Resq: Residual quantization for video perception," in *Proc. IEEE/CVF ICCV*, 2023, pp. 17 119–17 129.

[48] A. S. Parihar, D. Varshney, K. Pandya, and A. Aggarwal, "A comprehensive survey on video frame interpolation techniques," *The Visual Computer*, pp. 1–25, 2022.

[49] W. Bao, W.-S. Lai, C. Ma, X. Zhang, Z. Gao, and M.-H. Yang, "Depth-aware video frame interpolation," in *Proc. IEEE CVPR*, 2019, pp. 3703–3712.

[50] M. Usman, X. He, K.-M. Lam, M. Xu, S. M. M. Bokhari, and J. Chen, "Frame interpolation for cloud-based mobile video streaming," *IEEE Transactions on Multimedia*, vol. 18, no. 5, pp. 831–839, 2016.