

# RL-RC-DoT: A Block-level RL agent for Task-Aware Video Compression

Uri Gadot<sup>1,2</sup>   Assaf Shocher<sup>2</sup>   Shie Mannor<sup>1,2</sup>   Gal Chechik<sup>2</sup>   Assaf Hallak<sup>2</sup>

<sup>1</sup>Technion   <sup>2</sup>NVIDIA Research

Corresponding authors: uri.gadot@campus.technion.ac.il, ahallak@nvidia.com

## Abstract

*Video encoders optimize compression for human perception by minimizing reconstruction error under bit-rate constraints. In many modern applications such as autonomous driving, an overwhelming majority of videos serve as input for AI systems performing tasks like object recognition or segmentation, rather than being watched by humans. It is therefore useful to optimize the encoder for a downstream task instead of for perceptual image quality. However, a major challenge is how to combine such downstream optimization with existing standard video encoders, which are highly efficient and popular. Here, we address this challenge by controlling the Quantization Parameters (QPs) at the macro-block level to optimize the downstream task. This granular control allows us to prioritize encoding for task-relevant regions within each frame. We formulate this optimization problem as a Reinforcement Learning (RL) task, where the agent learns to balance long-term implications of choosing QPs on both task performance and bit-rate constraints. Notably, our policy does not require the downstream task as an input during inference, making it suitable for streaming applications and edge devices such as vehicles. We demonstrate significant improvements in two tasks, car detection, and ROI (saliency) encoding. Our approach improves task performance for a given bit rate compared to traditional task agnostic encoding methods, paving the way for more efficient task-aware video compression.*

## 1. Introduction

Video compression is an essential and widely studied problem [1, 3, 15, 32, 35]. Most video compression algorithms are designed for preserving how a video is perceived by people. With the success of computer vision applications, many videos are used in automated systems, from autonomous drones and cars, to security cameras, and in downstream tasks, like object detection or recognition. In these cases, compression must prioritize regions relevant to the task at hand (e.g., allocating more bits to objects than to

the background).

Real-world deployment of compression and computer vision systems complicates matters further. Video data must be collected in real time from devices, using low computational resources, and be usable for training various models across multiple tasks, not just for immediate inference. Furthermore, due to computational and hardware constraints, compression must be done without access to the ground truth for the downstream tasks during the encoding process. Our goal is to tackle these challenges by providing a general video compression method that can be adapted to any task, any encoding standard, operates in real-time, imposes low computational demands on the encoding side, and requires no ground-truth labels.

Two previous approaches were taken to this problem. The first relies on standardized video encoders [24, 32] which are highly efficient but are not designed for adapting compression to specific tasks in real-time. To address that, previous studies used downstream task information as input to the encoder side [17, 29, 39]. As one example, [39] performs semantic compression by applying a heavy feature extractor before encoding using a ground-truth segmentation maps. This approach may compress well at this setup, but typically require large computation resources before encoding, can not be used for various tasks, and is unsuitable for data collection. A second approach relies on deep encoding [21]. Once again it is computationally expensive and currently unsuitable for real-time applications or resource-constrained environments.

In this paper, we propose RL-RC-DoT, a novel solution to the problem of tuning an efficient real-time video compression system to a downstream task without access to its ground truth labels during inference. Our approach integrates a lightweight network on the video encoder side, trained to control the encoding process such that the decoded output is ideal for the task at hand. By leveraging standardized codecs, we ensure that our method is both computationally efficient and easily deployable across a range of devices. The solution allows for real-time video compression without requiring ground truth for downstream tasks. Our solution can be applied over any existing en-

coder, for simplicity, we chose to implement it over x264 as a mere example.

Coping with these challenges is hard. Standardized encoders are not differentiable, making it difficult to optimize bit allocation for specific tasks. To overcome this, we introduce a reinforcement learning (RL) mechanism that controls the Quantization Parameter (QP) at the macro-block (MB) level, adjusting the bit allocation for each block of the frame dynamically. This allows us to efficiently manage the bit-rate budget while optimizing task performance over an entire sequence of video frames. Our experiments demonstrate that this approach yields significant improvements in rate-distortion trade-offs, not just for the task the encoder was trained on, but also for other related tasks, showcasing the robustness of our method. Furthermore, we demonstrate its generalizability by showing how an encoder trained on one model can improve performance for other models without additional tuning.

In summary, this paper makes the following contributions. (1) We design the first task-aware video compression method that builds on top of existing encoders and does not require solving the task during inference. (2) We show how to optimize the quantization parameter of every MB in the frame while optimizing the performance of a downstream task on the reconstructed video under bit-rate constraints. (3) We design an architecture that outputs multiple actions, a tailored reward for this problem, and a task-prediction loss term. (4) We show improved rate-distortion trade-off for our agent on two tasks, car detection and ROI encoding with only small interference to image quality, and further show robustness to task shift, when tested on a related-but-different task than used for training.

## 1.1. Related Works

**Video compression with RL.** The integration of reinforcement learning into video compression has gained significant attention in recent years. Several approaches have focused on frame-level QP optimization [6, 11, 22, 23], using various techniques such as two-pass encoding and dual-critic architectures. While some researchers have proposed more granular control by developing QP selection agents at the Coding Tree Unit (CTU) or MacroBlock (MB) level [12, 13], these methods have been primarily validated only on image compression or intra-mode encoding scenarios.

**Task-aware video compression with unrestricted compute.** Several previous studies proposed video compression methods that are aware of a downstream task. [42] explored content-specific filters to improve post-processing in video codecs, optimizing them for machine vision tasks like object detection and segmentation. [10] introduced an encoder control for deep video compression that adapts to multiple tasks using a single pre-trained decoder, showing significant bit-rate improvement for object detection and

tracking. [30] highlighted the limitations of classical codecs in medical videos, proposing learned compression models to allocate more bits to medically relevant regions. [7] presented a semantic video encoding system that enhances object detection by selectively decompressing frames in surveillance streams. [16] developed a distributed compression framework that adjusts to varying bandwidth in multi-sensor networks to optimize task performance. [37] introduced an annotation-free optimization strategy that aligns video coding with machine tasks, improving rate savings without relying on ground truth data. Additionally, While [38] focused on real-time, quality-scalable video decoding, it also evaluated the codec performance on machine-based tasks.

All these approaches share a common limitation, they do not use the existing highly optimized and widely prevalent existing video compression ecosystem like the open-source x264 [24]. The challenge therefore remains to design video compression systems that build on top of existing technology, but can be tuned in a content adaptive way to a set of downstream tasks.

**Task-aware video compression with standard encoders** Another body of works does employ standardized encoders, but does not consider the inter-frame dependencies. [31] and [8] optimize the CTU partitioning to improve the compression for a downstream task. [9] uses a threshold on the saliency map to allocate more bits to important regions, while [4] optimizes over the modelled relation between each block parameter and the task performance. [17] uses RL for optimizing block QPs, but does so in each frame separately, where the sequence is defined over the sequence of blocks in the same frame. In our work we output all MB QPs with one policy and the sequence is defined over consecutive encoded frames in the video. The work most related to ours is [39], where they propose to use RL on both the QPs and block QPs in a hierarchical manner. However, they limit their optimization to only two frames in every GOP, and only two values of block QPs are chosen per block according to a given segmentation map. In our work we optimize over all frames and MB QPs, and we do not use any additional information like saliency, segmentation or downstream task during inference.

## 2. Preliminaries

### 2.1. Video Compression

Video compression is a process of reducing the size of digital video files while maintaining acceptable visual quality. It is a crucial technology in the modern digital age, enabling efficient storage, transmission, and streaming of video content across various platforms and devices. The primary goal of video compression is to eliminate redundant and less perceptible information from the video data according to con-

straints such as bit-rate of the target video.

One key aspect of video compression is the use of Quantization Parameters (QP). QP values control the level of compression and distortion applied to the video data, with higher values resulting in more compression but lower quality, and lower values preserving more detail but producing larger file sizes. In video encoding, QP can be applied at different levels of granularity. Frame QP refers to setting a single QP value for an entire frame, which is useful for maintaining consistent quality across the frame but may not be optimal for all areas. Per-block QP, conversely, allows for finer control by assigning different QP values to individual blocks within a frame, usually in small perturbations from a pre-assigned frame QP. This approach enables the encoder to apply more compression to less important or visually complex areas while preserving quality in critical regions. Per-MB QP can lead to more efficient compression and better overall visual quality, as it adapts to the local characteristics of the video content. It is especially suitable for task-aware optimization since most tasks target specific areas in the picture (e.g. object detection and segmentation).

One may wonder, if a downstream task is given, why is video compression needed at all? For instance, in the autonomous vehicle example, if a car detector is available, why not run that detector on the vehicle, and save only its decision instead of the compressed video. There are several strong reasons not to take this approach: (1) Many downstream tasks require resource-heavy networks that cannot run efficiently on-device, making it impractical to process the data locally. (2) Sending only task-specific features limits human interpretability, as there would be no watchable video for explainability. (3) This also confines the data to a single task, preventing its reuse for other applications or analyses. (4) Large-scale data collection, such as in autonomous driving, depends on compressed video storage; using features alone would limit future training and fine-tuning opportunities. (5) Task-specific features are often tied to a particular model, making them incompatible with new models, while compressed video remains adaptable across different systems. We show that our method allows different models to achieve high performance using the same compressed data. This is also the reason why we aim to develop a method that still preserves a video that would be meaningful to a person.

## 2.2. Reinforcement Learning

Reinforcement Learning (RL; [33]) is a field dealing with sequential decision making in unknown environments. To formulate a problem using RL, we first need to define its underlying Markov Decision Process (MDP). An MDP is defined by a tuple  $(\mathcal{S}, \mathcal{A}, P, R, \gamma)$ , where  $\mathcal{S}$  is a finite set of states,  $\mathcal{A}$  is a finite set of actions,  $P$  is a state transition probability function,  $P(s'|s, a)$ ,  $R$  is a reward function,  $R(s, a)$

and  $\gamma \in [0, 1]$  is a discount factor.

At each time step  $t$ , the agent observes the current state  $s_t \in \mathcal{S}$  and chooses an action  $a_t \in \mathcal{A}$ . The environment then transitions to a new state  $s_{t+1}$  with probability  $P(s_{t+1}|s_t, a_t)$  and the agent receives a reward  $r_t = R(s_t, a_t)$ . The goal of the agent is to find a policy  $\pi : \mathcal{S} \rightarrow \mathcal{A}$  that maximizes the expected cumulative discounted reward:

$$\max_{\pi} J^{\pi} = \mathbb{E}_{\pi, s_0 \sim \mu, s_{t+1} \sim P} \left[ \sum_{t=0}^{\infty} \gamma^t R(s_t, \pi(s_t)) \right]$$

To do so, many algorithms were proposed in the literature varying in their assumptions on the problem, computational complexity and data requirements. Perhaps the most widely used algorithm today is PPO [28] which directly optimizes the policy using full trajectories while constraining it from diverging.

When the action space is high-dimensional, as in our case, learning becomes exponentially harder. This can lead to extremely slow learning progress, and requires function approximators [34], which are prone to overfitting and generalize poorly. Consequently, techniques specifically designed for high-dimensional action spaces, such as action abstraction, or dimensionality reduction [26], are often necessary to make the problem tractable and enable efficient reinforcement learning.

## 3. Method

We present RL-RC-DoT, an **RL**-based **R**ate **C**ontroller for **D**ownstream **T**ask, that dynamically optimizes per-block QP deltas during video encoding. Simply put, our approach is to train a deep-RL agent that allocate bandwidth to different patches in each frame, only observing fast-to-compute features of every frame. To formalize the training framework, we cast the video compression problem with respect to a downstream task as a Markov decision process (MDP). In our formulation, each time-step of the MDP corresponds to processing one frame of the video. To train the policy, we use the PPO algorithm [28].

### 3.1. The MDP model

We now describe the various components of our model.

**State space:** We define the state of the environment to be per-patch statistics extracted from the encoder. The specific statistics we used are described below and in Appendix 7.1.1.

**Action space:** We define an action to be a choice of all MB QP deltas within a frame. For example, given a frame resolution of 480×320 pixels partitioned into 16×16 pixel blocks, the resulting action space constitutes a 30×20 dimensional matrix, where each value represent the delta between the frame level QP and the block specific QP. This

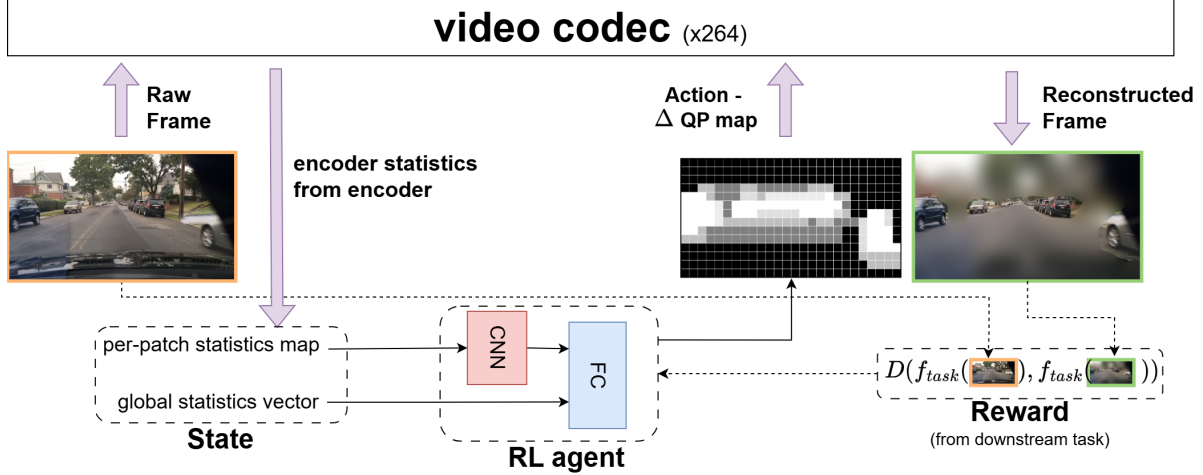


Figure 1. **RL-RC-DoT workflow.** Our proposed solution to the block-level control for a downstream task. RL-RC-DoT takes encoder statistics as input and outputs a block-level delta QP map. We then evaluate the difference in downstream task performance between the reconstructed frame and the raw frame. The reward contains both global score as reward and block-level score.

action space is very high-dimensional and we discuss below an efficiency improvement.

**Reward:** We define the reward as a combination of two rewards with different purposes.

First, we wish to maximize the performance of the downstream task on decoded frames. Since ground-truth data is unavailable, we introduce a novel self-supervised approach. This method treats the downstream task’s output on the original uncompressed frame as a pseudo-ground-truth, against which we evaluate the task performance on the reconstructed frame:

$$r_{\text{task}} = D[f_{\text{task}}(\text{frame}_{\text{raw}}), f_{\text{task}}(\text{frame}_{\text{recon}})],$$

where  $f_{\text{task}}$  is a pre-trained model for the downstream-task and  $D$  is a task-specific objective function. For example, for car detection,  $f_{\text{task}}$  we use a pre-trained car detection model (YOLO-v5 nano [14]), and  $D$  measures how well the downstream task model of  $f_{\text{task}}(\text{frame}_{\text{recon}})$  performs compared to the raw data  $f_{\text{task}}(\text{frame}_{\text{raw}})$ . Here,  $f_{\text{task}}(\text{frame}_{\text{raw}})$  is used as a pseudo-ground-truth, because we may not have access to the ground-truth of the frame.

The second component of the reward is designed for complying with the encoder’s bit-rate constraint. This is particularly crucial in streaming applications, where exceeding the allocated bandwidth can result in frame dropping and consequently deteriorate the viewer experience. The reward component  $r_{\text{bit-rate}}$  for this objective is:

$$r_{\text{bit-rate}} = -\left| \log \left( \frac{\text{current average bit-rate}}{\text{target bit-rate}} \right) \right|.$$

The final reward is a weighted combination of the two components:

$$r = r_{\text{bit-rate}} + \lambda r_{\text{task}},$$

where  $\lambda$  is a hyper-parameter that determines rate-performance trade-off.

**Efficient action space:** The naive action space described above is very high dimensional, presenting a significant computational challenge in term of convergence of reinforcement learning algorithms. To address this difficulty, we implement a coarsening approach: the agent operates on a lower-resolution action space, which is subsequently up-sampled to the original dimensions through interpolation. This technique facilitates more efficient training while maintaining the ability to generate fine-grained QP assignments. We analyze the impact of action space resolution on model performance in Appendix 7.5.

Finally, unlike previous approaches [17, 39] that relied on complex downstream task information (e.g. semantic maps, grad-cam), our method simplifies the state space by utilizing only encoder statistics that are readily available in the x264 encoder, so no additional calculations are required. A diagram of the full system is given in Figure 1 and additional details on the architecture used by the PPO algorithm is given in Appendix 7.1.2.

### 3.2. Macro-Block Reward Information

In most RL problems, the reward is a black-box directly mapping the state to a continuous score. Recent literature [40] has demonstrated that predictive modeling of rewards can significantly enhance agent performance. more specifically they implemented it as auxiliary heads alongside policy or value networks. In our setup, the reward presents a unique characteristic: the reward signals for various downstream tasks are often compositional. This occurs because the scores are derived from aggregating measure-

|           | Precision          |                    | Recall            |                   | PSNR        |             |
|-----------|--------------------|--------------------|-------------------|-------------------|-------------|-------------|
|           | low-rate           | high-rate          | low-rate          | high-rate         | low-rate    | high-rate   |
| x264      | .22 ± .0013        | .66 ± .0015        | .45 ± .002        | .81 ± .001        | 28.98 ± .03 | 34.55 ± .03 |
| RL-RC-DoT | <b>.36 ± .0015</b> | <b>.71 ± .0014</b> | <b>.63 ± .002</b> | <b>.83 ± .001</b> | 29.03 ± .03 | 34.55 ± .03 |

Table 1. Car detection precision and recall of YOLO5, and PSNR. Value are mean and s.e.m. calculated across all frames from the test set.

ments across local patches in each input frame. For example, when optimizing for saliency-weighted PSNR, the reward is computed by aggregating per-pixel reconstruction errors.

To leverage this decomposable nature of rewards, we propose to add an auxiliary prediction loss for these sub-scores during back-propagation. Specifically, we introduce a block-wise loss that aims to predict for each individual block, the local reward that contribute to the overall task score. Adding this loss for macro-block level reward information is expected to enhance the agent’s performance. First, it provides a more localized learning signal, allowing the agent to understand the impact of its actions on individual components of the reward. Second, by learning to predict these sub-scores, the agent develops a richer internal representation of the task structure. Lastly, this method aligns the agent’s learning more closely with the actual composition of the reward, potentially leading to faster convergence and more stable learning. Section 5.5 shows the effect of this improvement.

## 4. Experiments

We evaluate our approach with two downstream tasks: car detection and region-of-interest (ROI) encoding [19]. We further study the robustness of the method, when a trained compression policy is tested with a different car detector, or even in a segmentation task instead of detection. Finally, we report performance of ablation experiments.

### 4.1. Dataset

We trained and evaluated RL-RC-DoT using a subset of video streams from the BDD100K dataset [41], a large-scale driving video dataset, with multi-task annotations. We reconstructed the raw data from the videos and to allow faster training time, we resized them to a smaller resolution of 480x320 pixels. We then filtered out streams that exhibited trivial rate-task performance (RD) curves with respect to the downstream tasks of car detection precision, when encoded with the standard x264 codec [36]. We specifically excluded streams that showed zero precision across most target bit-rates. This approach ensured that our dataset presented meaningful challenges for compression optimization.

Our final dataset comprised of 172 streams in total, with 65 streams used for training our agent, 7 streams used for evaluation on different hyper-parameters and 100 streams

reserved for testing. For reproducibility, we provided a detailed list of the specific stream used in our experiments in Appendix 7.6 of this paper.

### 4.2. Evaluation metrics: RD-curve and BD-rate

Since compression is a constraint optimization problem, it is standard to depict results using a Rate-Distortion (RD) curve. Single RD-curve illustrates the trade-off between bit-rate constraint and quality in video compression (see Figure 2). RD-curves are traditionally used with PSNR, but are equally applicable to task-specific metrics like precision/recall for a detection task or saliency-weighted PSNR for ROI-based encoding. These RD-curves allow us to evaluate compression efficiency for any downstream tasks on reconstructed videos.

BD-rate (Bjontegaard Delta rate; [2]) is a widely used metric in video compression to compare the efficiency of different encoding methods. This method calculates the average difference in bit-rate between two rate-distortion (RD) curves at the same quality level. The BD-rate represents the percentage of bit-rate savings that one encoding method achieves over another while maintaining equivalent video quality performance. Thus, a negative BD-rate indicates that the test method requires less bits than the reference method to achieve the same quality / task performance.

### 4.3. Compared methods:

To conduct a fair and meaningful comparison against existing baselines, they should be solving the same task, and particularly have access to the same information. Several previous studies developed methods for task-aware encoding, but their setup is fundamentally different. For instance, some previous methods assume access to the downstream model during inference [39] resulting in a much higher computational cost depending on the task. [17] and [8] focus on a single-frame (image) compression, ignoring the overall video budget constraints. Finally, most methods did not release code [8, 29]. These differences in approach and constraints make direct comparisons misleading.

### 4.4. Experimental details

All of our experiments use the x264 open source encoder software [24], with the *medium* preset and target bit-rates 50 – 200 kbps. To extract the statistics we allow x264 to use look-ahead for 10 frames. For car detection, we employ YOLOv5-nano [14]. ROI encoding is evaluated using saliency maps generated by TranSalNet [20]. Our

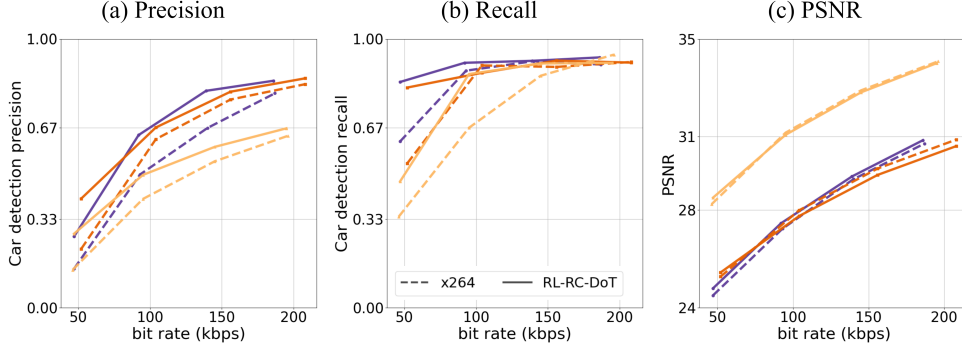


Figure 2. RD curves for Car detection task for different streams (color). Comparing standard x264 (dashed lines) with RL-RC-DoT (solid lines). Curves show 3 example streams, demonstrating how RL-RC-DoT improves quality across the range of bit-rate values. (a) Car detection precision (b) recall (c) PSNR.

agent is trained using Stable-Baselines3 [27] implemented PPO with the reward function described in Section 3. We augment the standard PPO algorithm with a reward per block prediction network, as described in Section 3.2. The computing time of RL-RC-DoT is 0.004 sec, equivalent to 250FPS. This is added on top of the compute time of the encoder. It ensures that our agent is suitable for real-time video compression.

## 5. Results

### 5.1. Car Detection

We first assess the performance of RL-RC-DoT in the context of video compression optimized for car detection. The reward function for training our RL agent is based on the precision score of YOLOv5-nano [14]. In practice it was calculated as true positives divided by total positives in the granularity of the detection boxes. For our additional auxiliary loss described in Section 3.2, we compute the precision score for each individual block separately to generate block-specific reward information.

Table 1 compares RL-RC-DoT with the standard x264 encoder, focusing on the detection performance of the YOLOv5-nano detector on compressed videos. The evaluation is conducted across multiple compression rates, with results averaged over all frames in the test dataset for each target bit-rate. We also applied the same comparative approach to assess the PSNR of the reconstructed streams. The results demonstrating that RL-RC-DoT improves car detection precision and recall significantly, with minimal impact on the PSNR of the compressed videos.

Figure 2 illustrates the superiority of RL-RC-DoT over the standard x264 encoder through RD curves for three representative video streams. Figure 3, shows a qualitative example of the performance gain. We compare the images in both types of rate-control, and the output of the downstream task. We can see the details corresponding to the downstream task are better reconstructed yielding a more relevant image.

To quantify the performance difference between meth-

ods, we compute the BD-rate (see 4.2), a standard metric in the field. Our approach shows significant improvements in detection performance, with BD-rate reductions of  $24.7\%(\pm 1.38\%)$ . These gains come at a minimal cost to overall video quality, yielding a slight increase (deterioration) in PSNR BD-rate of  $1.19\%(\pm 0.46\%)$  (Table 2). This means that videos compressed using RL-RC-DoT remains understandable to human viewers, a crucial aspect for validation and debug purposes. It also exhibits robustness to different task models, which we elaborate on in section 5.4.

### 5.2. Quantization map analysis

To gain more insight into how our approach affects QP map, we quantified the relation between the QP map and areas in the video that are useful for car detection. Specifically, we computed the KL-divergence between normalized QP-maps and Eigen-CAM [25] spatial map. Figure 4 illustrates the three maps for one frame, showing that RL-RC-DoT allocates more bits to areas in the frame that are informative for car detection. Figure 11 (provided in the supplementary material) aggregates this metric across our entire test dataset, confirming our approach’s effectiveness with significantly lower KL-divergence values (mean  $D_{kl}$  for RL-RC-DoT = 2.6 compared to x264=4.4).

### 5.3. ROI encoding

We conducted similar experiments for ROI-encoding task by promoting saliency weighted PSNR as the task score. RL-RC-DoT exhibits a BD-rate value of  $-25.64\%(\pm 0.99\%)$ , indicating that our method achieves significantly better quality in salient regions at lower bit-rates compared to x264. Interestingly, the PSNR BD-rate obtained is  $-5.26 \pm 0.36$  which is slightly better than the vanilla rate-control. This may be due to the proximity between the two tasks. This also shows the sub-optimality of the vanilla rate-control when considering specific content. Figure 5 illustrates the RD curves for three representative video streams. These curves demonstrate that in most cases, RL-RC-DoT achieves a more favorable RD trade-off for ROI encoding task compared to x264. Figure 6 provides

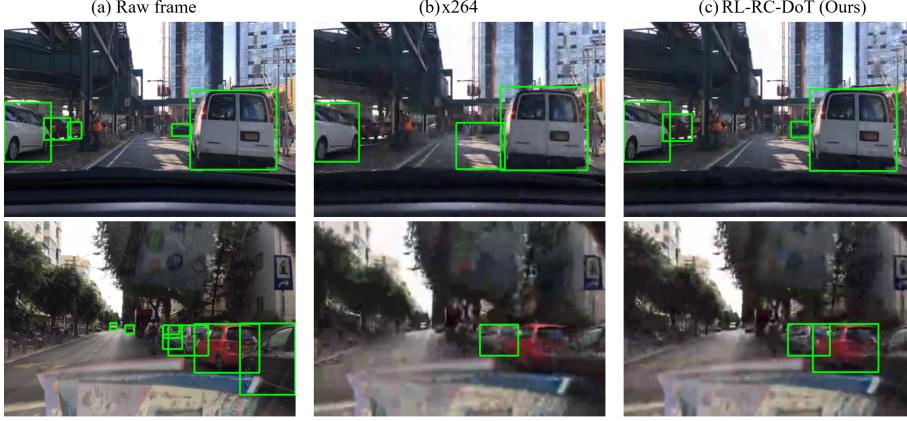


Figure 3. Car detection example result. (a) detection output on x264 reconstructed frame, (b) output on raw frame and (c) output on RL-RC-DoT reconstructed frame. Notice that both RL-RC-DoT and x264 used the same target bit-rate

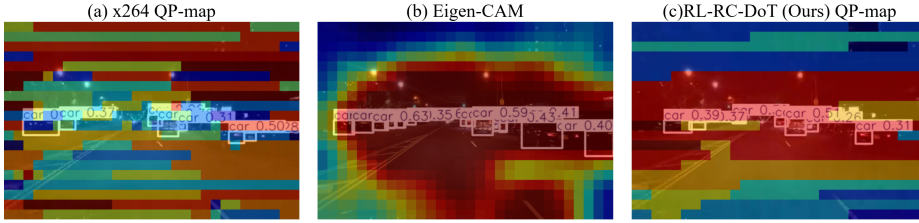


Figure 4. Example frame analysis. Red areas denote more bits allocate in the QP map ((a) x264 allocated bits, (c) RL-RC-DoT allocated bits) and (b) Eigen-CAM values (middle).

qualitative examples of our method’s performance, visually illustrating the enhanced quality in salient regions compared to the baseline encoding.

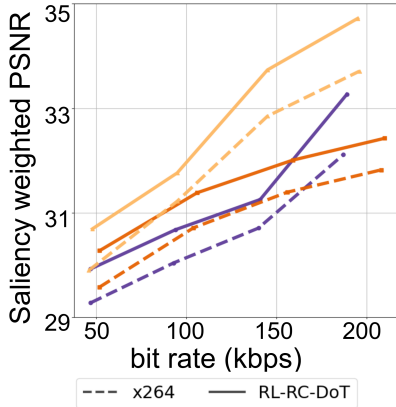


Figure 5. RD-curves for 3 videos (color) for ROI-encoding.

#### 5.4. Task Robustness

An important concern is that RL-RC-DoT might overfit for the training task. That would mean that changing the model, may harshly hurt performance. We set to evaluate robustness to such changes in RL-RC-DoT by training the policy with one downstream task, and testing it with another. More specifically, we optimized the policy for car detection using the YOLOv5-nano model, as described in section 5.1. Then, we measured the detection performance of another model, SSD [18]. We also measure the performance on the related

but distinct task of car segmentation (DeepLab; [5]). The results are also listed in Table 2.

| Precision (YOLO) | Recall (YOLO)     | PSNR             |
|------------------|-------------------|------------------|
| $-24.7 \pm 1.57$ | $-19.75 \pm 2.97$ | $1.19 \pm 0.46$  |
| Precision (SSD)  | Recall (SSD)      | Segmentation IOU |
| $-26.2 \pm 1.48$ | $-25.81 \pm 2.03$ | $-14.6 \pm 1.81$ |

Table 2. BD-rate Results on RL-RC-DoT applied on test set for the car detection task for various settings. Negative values mean that RL-RC-DoT improves over baseline.

This approach allows us to examine whether our method truly captures fundamental aspects of visual information relevant to automotive perception tasks, rather than overfitting to a specific model or narrow task definition. By demonstrating performance improvements across different models and related tasks, we aim to show that our compression method preserves task-relevant information in a more general sense, potentially allowing for model updates or task modifications without the need to retrain the compression policy. This robustness is crucial for real-world applications where deployed systems may need to adapt to new models or slightly different tasks over time.

For car detection evaluated with SSD, the precision BD-rate is very similar to precision with YOLOv5-nano, which was used for training. For car segmentation, although tested



Figure 6. Saliency weighted PSNR results. (a) x264 reconstructed frame, (b) Saliency map of raw frame, extracted with [20] (c) RL-RC-DoT reconstructed frame. Notice that both RL-RC-DoT and x264 used the same target bit-rate

with a different task, we still observe an improved but weaker BD-rate than the detection task. This improvement can be attributed to the close relation between the tasks, so meaningful macro-blocks for car detection, are also useful for the segmentation task. In summary, the BD-rate obtained on the PSNR and the various tasks show the robustness of our method to new tasks and new models that solve the task. Additional qualitative experiments examining task robustness are detailed in the appendix

| Car detection          | Precision<br>BD-rate | PSNR<br>BD-rate |
|------------------------|----------------------|-----------------|
| RL-RC-DoT              | $-24.7 \pm 1.57$     | $1.19 \pm 0.46$ |
| RL-RC-DoT w/o RI       | $-19.4 \pm 1.38$     | $1.92 \pm 0.41$ |
| RL-RC-DoT $\gamma = 0$ | $-9.78 \pm 1.29$     | $5.44 \pm 0.6$  |

| ROI encoding           | Sal.-weighted<br>PSNR BD-rate | PSNR<br>BD-rate  |
|------------------------|-------------------------------|------------------|
| RL-RC-DoT              | $-25.64 \pm 0.99$             | $-5.26 \pm 0.36$ |
| RL-RC-DoT w/o RI       | $-23.46 \pm 0.97$             | $-4.54 \pm 0.42$ |
| RL-RC-DoT $\gamma = 0$ | $-16.01 \pm 0.77$             | $2.11 \pm 0.31$  |

Table 3. Ablation study. (1) Full RL-RC-DoT (2) Omitting reward information (RI) from the training process and (3) Ignoring long term effects by using a myopic policy.

### 5.5. Ablation Experiments

To quantify the relative contribution of various components of our method, we perform ablation studies, for both car detection and ROI encoding, and provide the results in Table 3. For both tasks, we first ablated the macro-block reward information as described in Subsection 3.2. Then, ran an experiment for  $\gamma = 0$  which shows what happens when optimizing for a myopic policy.

The results show that reward info improved the learning process and reduces the BD-rate even further for both tasks. This demonstrates the benefit of exploiting additional infor-

mation in the video compression domain that is generally not available. For  $\gamma = 0$ , the BD-rate is significantly worse for both tasks. As expected, ignoring the future implications of the bit-allocation can cause sub-optimal decisions for the entire video. This also emphasizes the limitation of rate-control methods optimizing for every frame separately; a common practice by previous works.

## 6. Conclusions and Limitations

ML for video understanding is widespread but costly to store, making efficient compression essential. Current task-aware compression methods face limitations like computational demands and ground truth task data dependencies. We develop an efficient RL solution which encodes every frame in real time while optimizing the future bit-rate and task performance on the reconstructed video. Our learned policy is robust against changes in the downstream models for the same task and to closely related tasks, showing large important potential for data collection for autonomous vehicle, patient monitoring and robotics.

In this work, we focused on H.264 due to its widespread use, but our mechanism is inherently codec-agnostic. Extending it to H.265 or H.266 would require adaptations, such as incorporating encoder-specific features like CTU structures. Additionally, integrating unsupervised learning metrics into the reward function could improve task generalization by enabling the agent to learn more transferable representations of video quality. We propose these directions as promising avenues for future work.

**Limitations:** Training our models involves encoding and performing the downstream task per frame, and this may slow down converge depending on the complexity of the downstream task. Also, generalizing across video resolutions may be hard because it affects the size of action space and the complexity of the learning problem.

## References

- [1] Vasudev Bhaskaran and Konstantinos Konstantinides. Image and video compression standards: algorithms and architectures. 1997. 1
- [2] Gisle Bjøntegaard. Calculation of average psnr differences between rd-curves. 2001. 5
- [3] Benjamin Bross, Ye-Kui Wang, Yan Ye, Shan Liu, Jianle Chen, Gary J Sullivan, and Jens-Rainer Ohm. Overview of the versatile video coding (vvc) standard and its applications. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(10):3736–3764, 2021. 1
- [4] Qi Cai, Zhifeng Chen, Dapeng Oliver Wu, Shan Liu, and Xiang Li. A novel video coding strategy in hevc for object detection. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(12):4924–4937, 2021. 2
- [5] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017. 7
- [6] Lian-Ching Chen, Jun-Hao Hu, and Wen-Hsiao Peng. Reinforcement learning for hevc/h. 265 frame-level bit allocation. In *2018 IEEE 23rd International Conference on Digital Signal Processing (DSP)*, pages 1–5. IEEE, 2018. 2
- [7] Tarek Elgamal, Shu Shi, Varun Gupta, Rittwik Jana, and Klara Nahrstedt. Sieve: Semantically encoded video analytics on edge and cloud. In *2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS)*, pages 1383–1388. IEEE, 2020. 2
- [8] Kristian Fischer, Fabian Brand, Christian Herglotz, and André Kaup. Video coding for machines with feature-based rate-distortion optimization. In *2020 IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP)*, pages 1–6. IEEE, 2020. 2, 5
- [9] Leonardo Galteri, Marco Bertini, Lorenzo Seidenari, and Alberto Del Bimbo. Video compression for object detection algorithms. In *2018 24th International Conference on Pattern Recognition (ICPR)*, pages 3007–3012. IEEE, 2018. 2
- [10] Xingtong Ge, Jixiang Luo, Xinjie Zhang, Tongda Xu, Guo Lu, Dailan He, Jing Geng, Yan Wang, Jun Zhang, and Hongwei Qin. Task-aware encoder control for deep video compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26036–26045, 2024. 2
- [11] Yung-Han Ho, Guo-Lun Jin, Yun Liang, Wen-Hsiao Peng, and Xiaobo Li. A dual-critic reinforcement learning framework for frame-level bit allocation in hevc/h. 265. In *2021 Data compression conference (DCC)*, pages 13–22. IEEE, 2021. 2
- [12] Yung-Han Ho, Chia-Hao Kao, Wen-Hsiao Peng, and Ping-Chun Hsieh. Neural frank-wolfe policy optimization for region-of-interest intra-frame coding with hevc/h. 265. In *2022 IEEE International Conference on Visual Communications and Image Processing (VCIP)*, pages 1–5. IEEE, 2022. 2
- [13] Jun-Hao Hu, Wen-Hsiao Peng, and Chia-Hua Chung. Reinforcement learning for hevc/h. 265 intra-frame rate control. In *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–5. IEEE, 2018. 2
- [14] Glenn Jocher. ultralytics/yolov5. <https://github.com/ultralytics/ultralytics>, 2020. 4, 5, 6
- [15] Jan Kufa and Tomas Kratochvil. Software and hardware hevc encoding. In *2017 International Conference on Systems, Signals and Image Processing (IWSSIP)*, pages 1–5. IEEE, 2017. 1
- [16] Po-han Li, Sravan Kumar Ankireddy, Ruihan Philip Zhao, Hossein Nourkhiz Mahjoub, Ehsan Moradi Pari, Ufuk Topcu, Sandeep Chinchali, and Hyeji Kim. Task-aware distributed source coding under dynamic bandwidth. *Advances in Neural Information Processing Systems*, 36, 2024. 2
- [17] Xin Li, Jun Shi, and Zhibo Chen. Task-driven semantic coding via reinforcement learning. *IEEE Transactions on Image Processing*, 30:6307–6320, 2021. 1, 2, 4, 5
- [18] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *Computer Vision—ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 21–37. Springer, 2016. 7
- [19] Yang Liu, Zheng Guo Li, and Yeng Chai Soh. Region-of-interest based resource allocation for conversational video communication of h. 264/avc. *IEEE transactions on circuits and systems for video technology*, 18(1):134–139, 2008. 5
- [20] Jianxun Lou, Hanhe Lin, David Marshall, Dietmar Saupe, and Hantao Liu. Transanet: Towards perceptually relevant visual saliency prediction. *Neurocomputing*, 2022. 5, 8
- [21] Guo Lu, Wanli Ouyang, Dong Xu, Xiaoyun Zhang, Chunlei Cai, and Zhiyong Gao. Dvc: An end-to-end deep video compression framework. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11006–11015, 2019. 1
- [22] Amol Mandhane, Anton Zhernov, Maribeth Rauh, Chenjie Gu, Miaosen Wang, Flora Xue, Wendy Shang, Derek Pang, Rene Claus, Ching-Han Chiang, et al. Muzero with self-competition for rate control in vp9 video compression. *arXiv preprint arXiv:2202.06626*, 2022. 2
- [23] Hongzi Mao, Chenjie Gu, Miaosen Wang, Angie Chen, Nevena Lazic, Nir Levine, Derek Pang, Rene Claus, Marisabel Hechtman, Ching-Han Chiang, et al. Neural rate control for video encoding using imitation learning. *arXiv preprint arXiv:2012.05339*, 2020. 2
- [24] Loren Merritt and Rahul Vanam. x264: A high performance h. 264/avc encoder. [online\] http://neuron2.net/library/avc/overview\\_x264-v8-5.pdf](http://neuron2.net/library/avc/overview_x264-v8-5.pdf), 2006. 1, 2, 5
- [25] Mohammed Bany Muhammad and Mohammed Yeasin. Eigen-cam: Class activation map using principal components. In *2020 international joint conference on neural networks (IJCNN)*, pages 1–7. IEEE, 2020. 6
- [26] Thomas PIERROT, Valentin Macé, Jean-Baptiste Sevestre, Louis Monier, Alexandre Laterre, Nicolas Perrin, Karim Beguir, and Olivier Sigaud. Factored action spaces in deep reinforcement learning. 2021. 3
- [27] Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-

- baselines3: Reliable reinforcement learning implementations. Journal of Machine Learning Research, 22(268):1–8, 2021. [6](#)
- [28] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347, 2017. [3](#)
- [29] Jun Shi and Zhibo Chen. Reinforced bit allocation under task-driven semantic distortion metrics. In 2020 IEEE international symposium on circuits and systems (ISCAS), pages 1–5. IEEE, 2020. [1](#), [5](#)
- [30] Joel Shor and Nick Johnston. The need for medically aware video compression in gastroenterology. arXiv preprint arXiv:2211.01472, 2022. [2](#)
- [31] Pranee Singh, Edward J Delp, and Amy R Reibman. Video-analytics task-aware quad-tree partitioning and quantization for hevc. In 2022 IEEE International Conference on Image Processing (ICIP), pages 2936–2940. IEEE, 2022. [2](#)
- [32] Gary J Sullivan, Jens-Rainer Ohm, Woo-Jin Han, and Thomas Wiegand. Overview of the high efficiency video coding (hevc) standard. IEEE Transactions on circuits and systems for video technology, 22(12):1649–1668, 2012. [1](#)
- [33] Richard S. Sutton and Andrew G. Barto. Reinforcement Learning: An Introduction. The MIT Press, Cambridge, MA, 1998. [3](#)
- [34] Hado Van Hasselt and Marco A Wiering. Reinforcement learning in continuous action spaces. In 2007 IEEE International Symposium on Approximate Dynamic Programming and Reinforcement Learning, pages 272–279. IEEE, 2007. [3](#)
- [35] Stephan Wenger. H. 264/avc over ip. IEEE transactions on circuits and systems for video technology, 13(7):645–656, 2003. [1](#)
- [36] Thomas Wiegand, Gary J Sullivan, Gisle Bjontegaard, and Ajay Luthra. Overview of the h. 264/avc video coding standard. IEEE Transactions on circuits and systems for video technology, 13(7):560–576, 2003. [5](#)
- [37] Marc Windsheimer, Fabian Brand, and André Kaup. On annotation-free optimization of video coding for machines. arXiv preprint arXiv:2406.07938, 2024. [2](#)
- [38] Chang Wu, Guancheng Quan, Gang He, Xin-Quan Lai, Yunsong Li, Wenxin Yu, Xianmeng Lin, and Cheng Yang. Qs-nerv: Real-time quality-scalable decoding with neural representation for videos. In ACM Multimedia 2024, 2024. [2](#)
- [39] Guangqi Xie, Xin Li, Shiqi Lin, Zhibo Chen, Li Zhang, Kai Zhang, and Yue Li. Hierarchical reinforcement learning based video semantic coding for segmentation. In 2022 IEEE International Conference on Visual Communications and Image Processing (VCIP), pages 1–5. IEEE, 2022. [1](#), [2](#), [4](#), [5](#)
- [40] Weirui Ye, Shaohuai Liu, Thanard Kurutach, Pieter Abbeel, and Yang Gao. Mastering atari games with limited data. Advances in neural information processing systems, 34: 25476–25488, 2021. [4](#)
- [41] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 2636–2645, 2020. [5](#)
- [42] Honglei Zhang, Jukka I Ahonen, Nam Le, Ruiying Yang, and Francesco Cricri. Competitive learning for achieving content-specific filters in video coding for machines. arXiv preprint arXiv:2406.12367, 2024. [2](#)