Neural Frank-Wolfe Policy Optimization for Region-of-Interest Intra-Frame Coding with HEVC/H.265

***Abstract*—This paper presents a reinforcement learning (RL) framework that utilizes Frank-Wolfe policy optimization to solve Coding-Tree-Unit (CTU) bit allocation for Region-of-Interest (ROI) intra-frame coding. Most previous RL-based methods employ the single-critic design, where the rewards for distortion minimization and rate regularization are weighted by an empirically chosen hyper-parameter. Recently, the dual-critic design is proposed to update the actor by alternating the rate and distortion critics. However, its convergence is not guaranteed. To address these issues, we introduce Neural Frank-Wolfe Policy Optimization (NFWPO) in formulating the CTU-level bit allocation as an action-constrained RL problem. In this new framework, we exploit a rate critic to predict a feasible set of actions. With this feasible set, a distortion critic is invoked to update the actor to maximize the ROI-weighted image quality subject to a rate constraint. Experimental results produced with x265 confirm the superiority of the proposed method to the other baselines.**

***Index Terms*—Bit allocation, rate control, action-constrained reinforcement learning, region-of-interest (ROI)**

I. Introduction

Broadly, the task of bit allocation for intra-frame coding is to allocate bits to coding units at certain level in such a way that the reconstructed image quality is maximized subject to a rate constraint. In this paper, we tackle the bit allocation problem at the coding-tree-unit (CTU) level for intra-frame coding with HEVC/H.265. In particular, we consider to weight more heavily the reconstruction quality in regions of interest (ROI). Due to the rate constraint, the ROI prioritization, as well as coding dependencies between CTUs, this problem is in essence a dependent decision-making process.

Reinforcement learning (RL) lends itself to dependent decision-making. There have been several attempts at applying RL to address bit allocation and rate control for image/video coding. Among the prior works, the single-critic design is most popular. Chen *et al.* [1] and Zhou *et al.* [2] learn RL agents to determine frame-level quantization parameters (QP) for hierarchical B-frame coding and low-delay P-frame coding, respectively. Hu *et al.* [3] adopt a similar approach for CTU-level bit allocation, in addressing intra-frame rate control. Fu *et al.* [4] extend the idea to streaming applications. Ren *et al.* [5] tackles ROI-based coding by learning an RL agent for both frame-level and CTU-level bit allocation. These prior works utilize a single reward function usually having a form of $r_D+\lambda r_R$, where r_D and r_R are the distortion and rate rewards, respectively. The design, dubbed the single-critic method, trades off distortion minimization and rate regularization with a fixed hyper-parameter λ. However, it is difficult to choose a fixed λ value that can work well on various videos and bit rates.

Deviating from the single-critic approach, Ho *et al.* [6] learn two independent critics, one of which estimates r_D and the other r_R. They train the RL agent by alternatively using the distortion critic and the rate critic. Specifically, the distortion critic is utilized to update the agent when the rate constraint is satisfied; otherwise, the rate critic is used to train the agent to meet the rate constraint. Though avoiding the use of a fixed λ, the training convergence is not guaranteed for the dual-critic method.

In this paper, we propose an action-constrained RL framework via Neural Frank-Wolfe Policy Optimization (NFWPO) [7], aiming for ROI-based intra-frame coding. Similar to [6], our scheme is composed of a distortion critic and a rate critic. However, unlike [6], we apply the rate critic in specifying a state-dependent action feasible set. We then utilize NFWPO together with the distortion critic to identify within the feasible set an action that minimizes the ROI-weighted distortion. The action thus chosen serves as a target for training the agent. We stress that our work also differs from the vanilla NFWPO [7] in that our action feasible set is dynamically determined via the rate critic rather than predefined.

To the best of our knowledge, this work presents the first attempt at applying Neural Frank-Wolfe Policy Optimization to address bit allocation and rate control for image/video coding. We demonstrate its effectiveness by taking as an example CTU-level bit allocation for ROI-based intra-frame coding. Experimental results confirm its superiority to the single-critic and dual-critic methods.

II. Neural Frank-Wolfe Policy Optimization

NFWPO [7] is an action-constrained RL algorithm. The objective of the action-constrained RL is to maximize the reward-to-go $Q(s,a)$ subject to the feasible actions $\mathcal{C}(s)$:

$$\arg\max_{a\in\mathcal{C}(s)} Q(s,a), \tag{1}$$

where the reward-to-go $Q(s,a)$ is the expected cumulative future reward under the policy π. In this paper, the policy $\pi(s)$ is implemented by a continuous, deterministic actor network.

Some prior works [8] deal with the action-constrained RL by including a projection layer at the output of the actor

network. The projection layer projects the action onto the feasible set $\mathcal{C}(s)$ by

$$\prod\nolimits_{\mathcal{C}(s)}(a) = \arg\min_{y \in \mathcal{C}(s)} ||y - a||_2, \tag{2}$$

where $a = \pi(s)$ is the pre-projection action and $\prod_{\mathcal{C}(s)}(a)$ is the post-projection action. Maximizing $Q(s, \prod_{\mathcal{C}(s)}(a))$ through gradient ascent may run into the trouble of zero gradients. For example, consider constraining the action to be non-negative with Rectified Linear Unit (ReLU) as the projection layer. The zero-gradient issue occurs during back-propagation when the action falls in the negative region.

To avoid this issue, NFWPO updates the actor network in three consecutive steps. First, it identifies a feasible update direction $\bar{c}(s)$ according to

$$\bar{c}(s) = \arg\max_{c \in \mathcal{C}(s)} \langle c, \nabla_a Q(s,a)|_{a=\prod_{\mathcal{C}(s)}(\pi(s))} \rangle, \tag{3}$$

where the operator $\langle a, b \rangle$ takes the inner product of a and b. Second, it evaluates a reference action $\tilde{a}_s$ by

$$\tilde{a}_s = \prod\nolimits_{\mathcal{C}(s)}(\pi(s)) + \alpha(\bar{c}(s) - \prod\nolimits_{\mathcal{C}(s)}(\pi(s)), \tag{4}$$

where α is the learning rate of NFWPO. Lastly, it learns the actor network $\pi(s)$ through gradient decent by minimizing the squared error between the reference action $\tilde{a}_s$ and $\pi(s)$:

$$\mathcal{L}_{NFWPO} = (\pi(s) - \tilde{a}_s)^2. \tag{5}$$

The zero-gradient issue is circumvented during training as the projection layer is not involved in the back-propagation.

III. Proposed Method

The objective of the CTU-level bit allocation for ROI-based intra-frame coding is to minimize the ROI-weighted distortion subject to a rate constraint. This is achieved by choosing properly a quantization parameter (QP) for every CTU in an intra-frame. In symbols, we have

$$\arg\min_{\{QP_i\}} \sum_{i=1}^{N} D_i(QP_i) \text{ s.t.} \sum_{i=1}^{N} R_i(QP_i) \leq R_f, \tag{6}$$

where QP_i indicates the QP for the i-th CTU, N denotes the number of CTUs in a frame, $D_i(QP_i)$ is the distortion of CTU i encoded with QP_i, $R_i(QP_i)$ is the number of encoded bits of CTU i, and R_f is the frame-level bit budget. To achieve ROI-based intra-frame coding, the distortions of CTUs in ROI regions are weighted more heavily.

We formulate our task as an action-constrained RL problem. When determining QP_i to encode CTU i, we view the minimization of the cumulative distortion in Eq. (6) as the maximization of the reward-to-go in Eq. (1). To meet the rate constraint in Eq. (6), we limit QP_i to a state-dependent feasible set $\mathcal{C}(s_i)$, which is specified by a rate critic (Section III-D). Eq. (6) is then transformed into

$$\arg\max_{QP_i \in \mathcal{C}(s_i)} Q(s_i, QP_i), \forall i, \tag{7}$$

which takes the same form as Eq. (1) and allows us to train the actor with NFWPO (Section II).

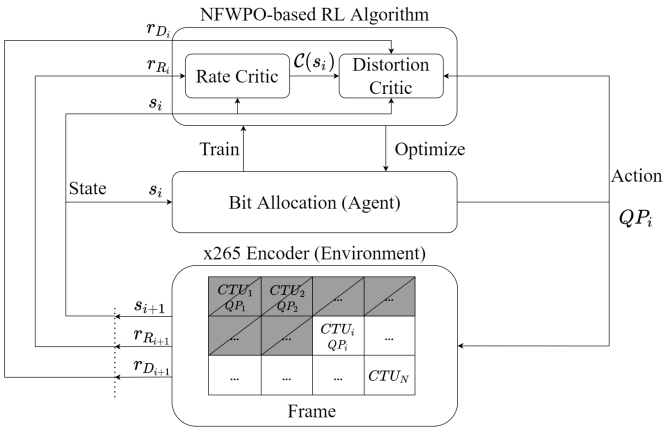

Fig. 1: The proposed NFWPO-based RL framework for CTU-level bit allocation.

TABLE I: State Definition

	Components
1	Variance of the current CTU
2	Gradient of the current CTU
3	Average of variances over remaining CTUs in the frame
4	Average of gradients over remaining CTUs in the frame
5	Percentage of the outstanding bits
6	Percentage of the remaining CTUs in the frame
7	Base QP (see Section III-B)
8	Bit budget of the current frame
9	Current CTU ROI indicator
10	Percentage of remaining ROI CTUs in the frame

A. System Overview

Fig. 1 illustrates our action-constrained RL framework. When encoding CTU i, a state s_i is first evaluated. Taking this state as input, our RL agent outputs an action QP_i (Section III-B). The x265 codec then encodes CTU i with QP_i. After encoding CTU i, we evaluate a distortion reward r_{D_i} and a rate reward r_{R_i} (Section III-C). These steps are repeated until all the CTUs in a frame are encoded.

At training time, the agent interacts with x265 by encoding every frame as an episodic task. We utilize the distortion and rate critics to predict the distortion reward-to-go $Q_D(s, QP)$ and the rate reward-to-go $Q_R(s, QP)$, respectively. The rate critic, which predicts the rate deviation from R_f at the end of encoding a frame, enables us to specify a feasible set $\mathcal{C}(s_i)$ of QP_i. The distortion critic, which estimates the cumulative distortion, guides the agent to minimize the total distortion. That is, Q_D will play the role of Q in Eq. (7).

B. States and Actions

Inspired by [3], we provide our (hand-crafted) state representation in Table I, which serves as the basis for the agent to output the action.

The action of our RL agent indicates the QP difference (also known as the delta QP) from the base QP. That is, the final QP value is the sum of the delta and the base QPs (i.e. $QP_i =$ delta QP $+$ base QP). The base QP is designed to reduce the search space of our agent; its value depends on the rate point.

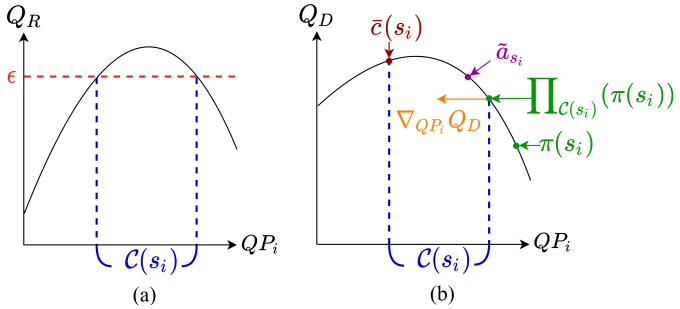

Fig. 2: Illustration of (a) the feasible set $\mathcal{C}(s)$, and (b) the reference action $\tilde{a}_s$.

C. Rewards

We specify two immediate rewards: the distortion reward r_{D_i} and the rate reward r_{R_i}. We define r_{D_i} as

$$r_{D_i} = \begin{cases} -\mathrm{D}_i \cdot w, & \text{if CTU}_i \text{ is ROI;} \\ -\mathrm{D}_i, & \text{otherwise,} \end{cases} \tag{8}$$

where D_i is the mean-squared error (MSE) of CTU i when encoded with the QP value chosen by the agent, and $w \geq 1$ is used to weight more heavily the distortions of ROI CTUs.

To construct the feasible set that addresses the rate constraint, the immediate rate reward r_{R_i} is designed as

$$r_{R_i} = \begin{cases} \frac{-|R_f - \sum_{t=1}^{N} R_t(QP_t)|}{R_f}, & \text{if } i = N; \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

The $\sum_{i=1}^{N} r_{R_i}$ represents the negative absolute deviation of the coding bit rate from the target R_f in percentage terms.

With r_{D_i} and r_{R_i}, the distortion and rate reward-to-go's $Q_D(s, QP)$ and $Q_R(s, QP)$ in Section III-A are given by

$$Q_D(s_i, QP_i) = E_{(s_t, QP_t)\sim\pi}[\sum_{t=i}^{N} \gamma^{t-i} r_{D_t}] \tag{10}$$

$$Q_R(s_i, QP_i) = E_{(s_t, QP_t)\sim\pi}[\sum_{t=i}^{N} \gamma^{t-i} r_{R_t}], \tag{11}$$

where γ is the discount factor. Particularly, these reward-to-go functions are approximated by the distortion and rate critics.

D. NFWPO-based RL for CTU-level Bit Allocation

This section presents how we use the two critic networks to implement NFWPO. First, we identify the feasible set $\mathcal{C}(s_i)$ for CTU i by the rate critic. To satisfy R_f, $\mathcal{C}(s_i)$ includes the QP values QP_i that the rate reward-to-go $Q_R(s_i, QP_i)$ is greater than or equal to a threshold ϵ (see Fig. 2 (a)):

$$\mathcal{C}(s_i) = \{QP_i | Q_R(s_i, QP_i) \geq \epsilon\}. \tag{12}$$

According to Eqs. (9) and (11), $\mathcal{C}(s_i)$ contains QP values that ensure the absolute rate deviation is capped by ϵ. Specifically, we discretize these QP values by querying the rate critic Q_R with a discrete step of 0.1 in the range of delta QP (i.e. $QP_i = \{\text{base QP} - 10, \text{base QP} - 9.9, ..., \text{base QP} + 10\}$).

Given the feasible set $\mathcal{C}(s_i)$, we follow the procedure in Section II to generate the reference action $\tilde{a}_s$ (Fig. 2 (b)). If the actor output $\pi(s_i)$ is outside of the feasible set, it will be projected onto the feasible set to reach $\Pi_{\mathcal{C}(s_i)}(\pi(s_i))$. We then derive $\bar{c}(s_i)$ based on Eq. (3), where we replace $Q(s, a)$ with $Q_D(s, a)$, and get $\tilde{a}_s$ by Eq. (4). Finally, we update the actor network via Eq. (5). Algorithm 1 summarizes the proposed method.

Algorithm 1 The proposed NFWPO-based RL algorithm

Randomly initialize critics $Q_D(s, a|w_D), Q_R(s, a|w_R)$, and actor $\pi(s|\theta)$ with weights w_D, w_R, and θ
Initialize target networks $Q'_D(s, a|w'_D), Q'_R(s, a|w'_R)$, and $\pi'(s|\theta')$ with weights $w'_D \leftarrow w_D, w'_R \leftarrow w_R$, and $\theta' \leftarrow \theta$
Initialize replay buffer R
for episode = 1 to M **do**
  Initialize a random noise process $\mathcal{N}$ for action exploration
  Evaluate initial state s_1
  for CTU $i = 1$ to N in a frame **do**
    Set $a_i = \pi(s_i|\theta) + \mathcal{N}_i$
    Encode CTU i with $QP = a_i$
    Evaluate the immediate rewards r_{D_i}, r_{R_i} and the new state s_{i+1}
    Store transition $(s_i, a_i, r_{D_i}, r_{R_i}, s_{i+1})$ in R
  end for
  Sample $\mathcal{B}$ transitions $(s_b, a_b, r_{D_b}, r_{R_b}, s_{b+1})$ from R
  Set $y_{D_b} = r_{D_b} + \gamma Q'_D(s_{b+1}, \pi'(s_{b+1}|\theta')|w'_D)$
  Update Q_D by minimizing $\mathcal{L} = \frac{1}{\mathcal{B}} \Sigma_b (y_{D_b} - Q_D(s_b, a_b|w_D))^2$
  Set $y_{R_b} = r_{R_b} + \gamma Q'_R(s_{b+1}, \pi'(s_{b+1}|\theta')|w'_R)$
  Update Q_R by minimizing $\mathcal{L} = \frac{1}{\mathcal{B}} \Sigma_b (y_{R_b} - Q_R(s_b, a_b|w_R))^2$
  for each state s $\in \mathcal{B}$ **do**
    Identify the feasible set $\mathcal{C}(s)$ by Eq. (12)
    Obtain $\bar{c}(s)$ by replacing Q with Q_D in Eq. (3)
    Obtain the reference action $\tilde{a}_s$ by Eq. (4)
    Update the actor network π by Eq. (5)
  end for
  Update target networks Q'_D, Q'_R,and π'
end for

IV. Experimental Results

A. Settings and Training Details

We conduct our experiments using x265 under the all-intra configuration (*--keyint 1*). All the sequences in our experiments are resized to 512×320 with CTU size 64×64, resulting in 40 CTUs per frame. The experiments follow the same frame-level bit allocation as x265. That is, we encode every sequence with fixed QPs 22, 27, 32, and 37 to establish the frame-level bit budget R_f, and turning on *--tune psnr* to optimize for PSNR.

Two datasets are used in our experiments, DAVIS 2017 TrainVal [9] and COCO 2017 validation [10], both of which provide ground truth object masks. For training, we use 64 sequences from DAVIS, and the number and position of ROI CTUs are sampled randomly.

At test time, we experiment with 4 settings on 3 video test sets. We utilize the ground truth masks to specify CTU-level ROI by defining ROI as CTUs that overlap with the selected object mask. In Table II, the *regular ROI* setting is tested on ***DAVIS*** test set (#1), which consists of 20 sequences from DAVIS dataset with ROI specified by the ground truth object masks, and on ***COCO1*** (#2), formed by 1600 images from COCO dataset with ROI given by the object masks of randomly chosen categories. We experiment with the *small ROI* and the *large ROI* settings on ***COCO2*** test set (#3, 4), respectively. The ***COCO2*** test set is composed of 950 images collected from COCO dataset. The *small ROI* setting is tested on the selected images, where ROI corresponds to the smallest object that covers no more than 5 CTUs. The *large ROI* setting simply inverts the ROI specification of the *small ROI* setting.

TABLE II: Comparison of Rate Deviations (at the lowest rate point) and BD-rates in different ROI settings (x265 as anchor)

#	Test Set	ROI Condition	Rate Deviation (%)			BD-Rate (%)			ROI Details	Avg. ROI size (CTU)
			single	Dual	Ours	single	Dual	Ours		
1	***DAVIS***	Regular	6.39	1.57	**0.96**	**-20.09**	-15.98	-18.79	CTUs corresponding to all objects in mask	10.7
2	***COCO1***	Regular	19.95	4.92	**3.26**	-6.31	-6.53	**-6.63**	CTUs corresponding to randomly picked objects	15.9
3	***COCO2***	Small ROI	4.59	2.45	**1.50**	-2.92	-5.51	**-7.34**	CTUs corresponding to the smallest object	2.5
4		Large ROI	48.59	6.92	**5.91**	6.42	6.47	**5.21**	Inversion of Small ROI	37.5

To train the actor and critic networks, we choose $\alpha = 0.05$ in Eq. (4), the ROI weighting parameter $w = 10$ in Eq. (8), the learning rate to be 0.001, and the 3-step temporal difference method. The base QPs are set to $QP_l - 3$, where QP_l are 22, 27, 32, and 37. The delta QP ranges from -10 to 10. The threshold ϵ in Eq. (12) is set to -0.05, allowing for a maximum rate deviation of $\pm 5\%$. For a fair comparison, the same rate tolerance is applied to the single- and dual-critic methods.

B. Rate-distortion Performance and Rate Deviations

Table II presents BD-rates (in terms of PSNR-YUV where the Y,U,V distortions are weighted in proportional to 6:1:1) and rate deviations, with x265 serving as anchor. In particular, the ROI-weighted MSE is evaluated according to

$$\text{ROI-Weighted MSE} = \frac{\text{MSE}_{\text{ROI}} \times 10 + \text{MSE}_{\text{NROI}}}{N_{\text{ROI}} \times 10 + N_{\text{NROI}}}, \quad (13)$$

where MSE_{ROI} and N_{ROI} are the sum of MSEs and the number of ROI CTUs, respectively; MSE_{NROI} and N_{NROI} are those of non-ROI CTUs. When reporting the average absolute rate deviations from the frame-level bit budget R_f, any deviation within $\pm 5\%$ of the R_f is regarded as 0% to reflect our $\pm 5\%$ tolerance.

From Table II, we see that our scheme achieves the smallest rate deviation under all the ROI settings, compared with the single-critic [3] and dual-critic [6] methods. In contrast, the rate deviation of the single-critic method is seen to be as high as 50% under the *Large ROI* setting. This is attributed to the use of a fixed λ for combining the distortion and the rate rewards ($r_D + \lambda r_R$). In the present case, the cumulative distortion reward may change drastically with the number of ROI CTUs (cp. Eq. (8)). This makes it difficult to identify a fixed λ that can work well under various ROI settings. We observe that the policy learned by the single-critic method [3] achieves the best test result only when the number of ROI CTUs is close to the training average (i.e. 20) and the test sequences share similar characteristics to the training data (i.e. DAVIS). When tested with *large ROI*, the single-critic method chooses too low a QP for ROI CTUs, leading to large rate overshooting (see #2, 4); in the other extreme with *small ROI*, it assigns too high a QP to non-ROI CTUs, resulting in poor rate-distortion performance (#3). In terms of BD-rate performance, our method outperforms the single-critic [3] and the dual-critic [6] methods in most of the settings. Even though the single-critic method shows slightly better BD-rate results in setting #1 (where the ROI setting and sequences are more similar to those used for training), it exhibits poor performance in the other settings (e.g. #3, 4), which underlines its poor generalization performance. In comparison, our method shows more consistent results across different settings.

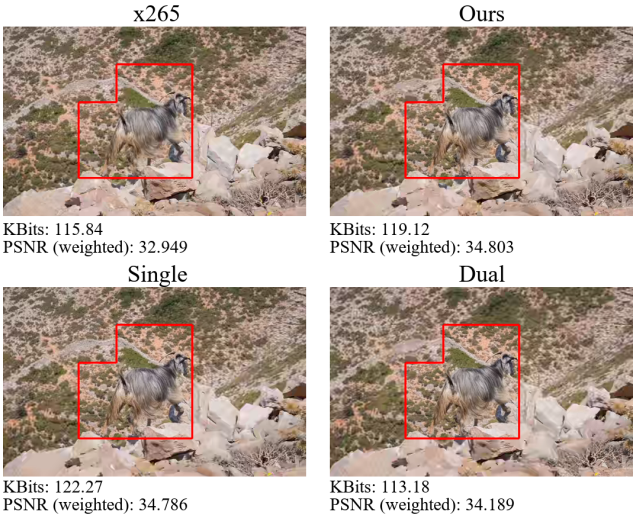

Fig. 3: Subjective quality comparison with ROI highlighted.

Fig. 4: Visualization of QP assignment.

Fig. 3 further presents a subjective quality comparison. As compared to the other methods, ours preserves more texture details in ROI and shows less blocking artifacts. Fig. 4 visualizes the corresponding QP assignment. Our method assigns lower QPs in ROI CTUs, which is in stark contrast to x265. One thing to note is that both the dual-critic method and ours choose a low QP for the last non-ROI CTU. This is resulted from the higher QPs assigned to previous CTUs. To meet the rate constraint, a lower QP is chosen for the last CTU.

V. Conclusion

This paper introduces a NFWPO-based RL framework for ROI-based intra-frame coding with HEVC/H.265. It overcomes the empirical choice of the hyper-parameter in the single-critic method and the convergence issue of the dual-critic method. It outperforms these two baselines, demonstrating better ability to generalize to various ROI settings.

REFERENCES

[1] L.-C. Chen, J.-H. Hu, and W.-H. Peng, "Reinforcement learning for HEVC/H.265 frame-level bit allocation," in *2018 IEEE 23rd International Conference on Digital Signal Processing (DSP)*, 2018, pp. 1–5.

[2] M. Zhou, X. Wei, S. Kwong, W. Jia, and B. Fang, "Rate control method based on deep reinforcement learning for dynamic video sequences in HEVC," *IEEE Transactions on Multimedia*, 2020.

[3] J.-H. Hu, W.-H. Peng, and C.-H. Chung, "Reinforcement learning for HEVC/H.265 intra-frame rate control," in *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2018.

[4] J. Fu, C. Hou, and Z. Chen, "360hrl: Hierarchical reinforcement learning based rate adaptation for 360-degree video streaming," in *2021 International Conference on Visual Communications and Image Processing (VCIP)*, 2021, pp. 1–5.

[5] G. Ren, Z. Liu, Z. Chen, and S. Liu, "Reinforcement learning based ROI bit allocation for gaming video coding in VVC," in *2021 International Conference on Visual Communications and Image Processing (VCIP)*, 2021, pp. 1–5.

[6] Y.-H. Ho, G.-L. Jin, Y. Liang, W.-H. Peng, and X. Li, "A dual-critic reinforcement learning framework for frame-level bit allocation in HEVC/H.265," in *2021 Data Compression Conference (DCC)*, 2021, pp. 13–22.

[7] J.-L. Lin, W. Hung, S.-H. Yang, P.-C. Hsieh, and X. Liu, "Escaping from zero gradient: Revisiting action-constrained reinforcement learning via Frank-Wolfe policy optimization," *2021 The Conference on Uncertainty in Artificial Intelligence (UAI)*, 2021.

[8] G. Dalal, K. Dvijotham, M. Vecerik, T. Hester, C. Paduraru, and Y. Tassa, "Safe exploration in continuous action spaces," *arXiv preprint arXiv:1801.08757*, 2018.

[9] J. Pont-Tuset, F. Perazzi, S. Caelles, P. Arbeláez, A. Sorkine-Hornung, and L. Van Gool, "The 2017 davis challenge on video object segmentation," *arXiv preprint arXiv:1704.00675*, 2017.

[10] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.