Optimized Bitrate Ladders for Adaptive Video Streaming with Deep Reinforcement Learning

Tianchi Huang[1], Lifeng Sun[1,2,3*]
[1]Dept. of CS & Tech., [2]BNRist, Tsinghua University. [3]Key Laboratory of Pervasive Computing, China

ABSTRACT

In the adaptive video streaming scenario, videos are pre-chunked and pre-encoded according to a set of resolution-bitrate/quality pairs on the server-side, namely *bitrate ladder*. Hence, we propose *DeepLadder*, which adopts state-of-the-art deep reinforcement learning (DRL) method to optimize the bitrate ladder by considering video content features, current network capacities, as well as the storage cost. Experimental results on both Constant Bitrate (CBR) and Variable Bitrate (VBR)-encoded videos demonstrate that DeepLadder significantly improvements on average video quality, bandwidth utilization, and storage overhead in comparison to prior work.

CCS CONCEPTS

• **Information systems** → *Multimedia streaming*; • **Computing methodologies** → *Neural networks*;

KEYWORDS

Bitrate Ladder Optimization, Deep Reinforcement Learning.

ACM Reference Format:
Tianchi Huang, Lifeng Sun. 2020. Optimized Bitrate Ladders for Adaptive Video Streaming with Deep Reinforcement Learning. In *Annual conference of the ACM Special Interest Group on Data Communication on the applications, technologies, architectures, and protocols for computer communication (SIGCOMM '20 Demos and Posters), August 10–14, 2020, Virtual Event, USA*. ACM, New York, NY, USA, Article 4, 3 pages. https://doi.org/10.1145/3405837.3411387

1 INTRODUCTION

The Global Internet Phenomena Report COVID-19 Spotlight [9] reveals that at least 51.43% of the video streams are delivered by adaptive video streaming technologies. In such technologies, on the server-side, video streams are pre-chunked and pre-encoded at various bitrates or quality levels. On the client-side, users often adopt adaptive bitrate (ABR) algorithms to pick the proper bitrates for chunks to ensure the quality of experience (QoE). We often called the server-side encoding settings as *bitrate ladder* [1].

The majority of existing bitrate ladder optimization schemes often use fixed encoding ladders (known as *one-size-fits-all* [3]), or only consider video contents [2] or network conditions [7], and solve the problem mathematically. In this poster, we propose DeepLadder, a per-chunk video transcoding system. Technically, we set video contents, current network traffic distributions, past actions as the state, and utilize a neural network (NN) to determine the proper action for each resolution *autoregressively*. Unlike the traditional bitrate ladder method that outputs all candidates at one step, we model the optimization process as a Markov Decision Process (MDP), which can effectively reduce the action space. DeepLadder employs Dual-PPO [12], the state-of-the-art deep reinforcement learning method, to learn the policy by interacting with the environments without any presumptions, attempting to balance a variety of goals such as maximizing perceptual video quality, improving bandwidth utilization, and reducing the storage overhead.

We implement DeepLadder-CBR and DeepLadder-VBR for dealing videos with different encoding types (i.e. constant bitrate (CBR)-encoded and variable bitrate(VBR)-encoded videos). Using trace-driven experiments, first, comparing DeepLadder-CBR with existing methods under various networks and videos, we observe that DeepLadder improves the average quality by 8.49%-14.25%, enhances the bandwidth utilization by 3.66%-11.68%, and reduces the storage cost by 39.77%-54.84%. Further, the results on DeepLadder-VBR also illustrate the significant improvements in the average quality (3.8%-7.7%) and the storage costs (21.26%-44.88%). Finally, we demonstrate that DeepLadder is practical and can be deployed into the transcoding framework. Testing results on both CPU and GPU devices indicate that DeepLadder achieves the aforementioned performance with only 3.4% on extra costs.

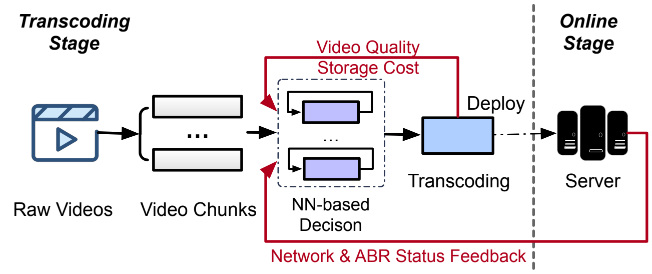

Figure 1: An Overview of DeepLadder's System. We leverage a NN-based decision model for constructing the proper bitrate ladders, and transcode the video according to the assigned settings.

2 SYSTEM MECHANISM

Overview. As illustrated in Figure 1, DeepLadder's basic system work-flow is mainly composed of the transcoding stage and the online stage. In the transcoding stage, we leverage an NN-based decision model for constructing the proper bitrate ladders and use

* Corresponding Author. *{htc19@mails, sunlf@}tsinghua.edu.cn.*

SIGCOMM '20 Demos and Posters, August 10–14, 2020, Virtual Event, USA

ACM ISBN 978-1-4503-8048-5/20/08…$15.00
https://doi.org/10.1145/3405837.3411387

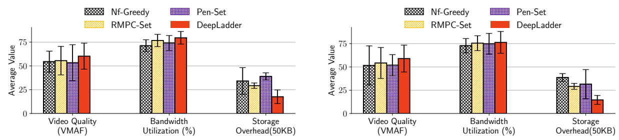

Figure 2: DeepLadder-CBR vs. existing approaches.

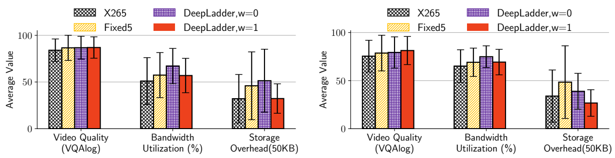

Figure 3: DeepLadder-VBR vs. basselines.

the assigned settings to transcode the video for each chunk. In the online stage, the encoded chunks are deployed on the server and wait for fetching. The NN instantly obtains the video quality and storage from the transcoding stage, as well as periodically receives the network status feedback from the online stage.

Details. We mainly categorize the DeepLadder's state input into four parts, that is, current network traffic distributions, video features, past actions, and desired reward weights w. DeepLadder is allowed to support diversity of coding methods. For the CBR-encoded videos, we set the the action $a_i = (N_{i-1} + N_i)/2$, where N_i represents the i-th throughput categorized by the bandwidth histogram method. For the VBR-encoded videos, we set the action size of DeepLadder-VBR as 20, symbolizing that the videos will be encoded by Constant Rate Factor (CRF)= $20 - 39$. Such settings have already covered the best quality for most videos. The reward function is listed in Eq. 1, given a network condition C, we aim to maximize the bandwidth utilization, the average video quality and minimize the storage cost. Here U is the bandwidth utilization, Q is the average video quality and Sz is the storage cost, w is a scalar which controls the importance of the storage metric. Moreover, we use Video Multimethod Assessment Fusion (VMAF) [6] as the quality metrics and measure the quality metric for each video chunk with all the resolution and encoding bitrate tracks.

$$r_t = \sum_t U(a_t, C) + \sum_t Q(a_t) - w \sum_t Sz(at)/t. \quad (1)$$

3 PRELIMINARY RESULTS

Experimental Setup. To better simulate the diversity of real-world network conditions, we collect over 3,000 network traces, totally almost 50 hours, from public datasets for training and testing. For the CBR-encoded video scenario, we list the baselines as:

▹ **Netflix [3] (Nf-Greedy):** offline picking the results of all the bitrate-resolutions pairs for each resolution, and select the best pair which is close to the convex hull as possible. We add the algorithm into the baseline since it's scalable, simple yet effective.

▹ **Pensieve-Settings [4] (Pen-Set):** a popular fixed encoding ladder profile, which encodes the videos at the bitrate of {300, 750, 1200, 1850, 2850, 4300}kbps.

Table 1: Real-world implementation of DeepLadder.

Process	Video Content Extraction	Inference Timespan	Transcoding Time	Extra Cost (%)
CPU(ms)	439	3	12060	**3.67**
GPU(ms)	94	12	3407	**3.11**

▹ **RobustMPC-Settings [13] (RMPC-Set):** an another fixed encoding ladder settings with {300, 700, 1000, 1500, 2000, 3000}kbps.

Furthermore, we select the following baselines for the VBR-encoded video scenario:

▹ **Fixed-25 [5]:** recent studies find that using the CRF value of 25 can provide good viewing quality.

▹ **X265 [11]:** the default CRF is fixed as 28 [8].

Trace-driven Evaluation. As shown in Figure 2, we test the performance of each scheme by the video under the HSDPA and FCC network environment and report the average video quality, bandwidth utilization and storage overhead. We can see that DeepLadder outperforms previously proposed methods, with the improvements on i) average video quality of 8.49% - 12.94% on the HSDPA dataset and 8.78%-14.25% on the FCC dataset; ii) 3.66%-11.68% in terms of the bandwidth utilization on HSDPA dataset. Meanwhile, DeepLadder reduces the storage overhead by 39.77%-54.84% compared with the baselines. Figure 3 illustrates the results for the Oboe and HDFS network conditions. As expected, DeepLadder-VBR ($w = 0$)'s bandwidth utilization is within 8.37%-15.03% compared with fixed encoding ladder schemes. Moreover, DeepLadder-VBR ($w = 1$) effectively *preserves* the average video quality and reduces storage cost by 29.8% on the Oboe traces, 21.26%-44.88% on the HDFS network.

Practical Implementation. We deploy the proposed scheme as a service into the transcoding frameworks and list the time consumption for each process in Table 1, tested on an Intel Core i7 CPU2.2GHz and an Nvidia 1080Ti GPU. In particular, in the GPU test, we use the FFmpeg's integration of NVIDIA Video Codec SDK that enables high-performance hardware-accelerated video pipelines [10]. We show that DeepLadder (in gray) only consumes 3.11%-3.67% on the extra cost compared with the original system.

4 SUMMARY AND FUTURE WORK

We present DeepLadder, a novel bitrate ladder construction system which utilizes state-of-the-art NN-based model to build the proper bitrate encoding profile. DeepLadder's NN takes current raw video contents, network traffic capacities, and the storage overhead into the input. We adopt state-of-the-art DRL method to train DeepLadder w.r.t a large corpus of internet network traces and collected videos. Experimental results on both CBR and VBR-encoded videos indicate the superiority of DeepLadder over existing approaches. Eventually, DeepLadder sheds a light on how to optimize the transcoding pipeline with a smart yet practical manner.

Our additional research may focus on: i) consider the performance of different off-the-shelf ABR algorithms; ii) enhance the scalability of DeepLadder; iii) consider supporting both per-chunk and per-title encoding services.

Acknowledgement. We thank the *SIGCOMM* reviewer for the valuable feedback. This work was supported by NSFC under Grant 61936011, 61521002, Beijing Key Lab of Networked Multimedia, and National Key R&D Program of China (No. 2018YFB1003703).

REFERENCES

[1] Abdelhak Bentaleb, Bayan Taani, Ali C Begen, Christian Timmerer, and Roger Zimmermann. 2018. A Survey on Bitrate Adaptation Schemes for Streaming Media over HTTP. *IEEE Communications Surveys & Tutorials* (2018).

[2] Chao Chen, Yao-Chung Lin, Steve Benting, and Anil Kokaram. 2018. Optimized transcoding for large scale adaptive streaming using playback statistics. In *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, 3269–3273.

[3] Jan De Cock, Zhi Li, Megha Manohara, and Anne Aaron. 2016. Complexity-based consistent-quality encoding in the cloud. In *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE, 1484–1488.

[4] Hongzi Mao, Ravi Netravali, and Mohammad Alizadeh. 2017. Neural adaptive video streaming with pensieve. In *SIGCOMM 2017*. ACM, 197–210.

[5] Yanyuan Qin, Shuai Hao, Krishna R Pattipati, Feng Qian, Subhabrata Sen, Bing Wang, and Chaoqun Yue. 2018. ABR streaming of VBR-encoded videos: characterization, challenges, and solutions. In *Proceedings of the 14th International Conference on emerging Networking EXperiments and Technologies*. 366–378.

[6] Reza Rassool. 2017. VMAF reproducibility: Validating a perceptual practical video quality metric. In *Broadband Multimedia Systems and Broadcasting (BMSB), 2017 IEEE International Symposium on*. IEEE, 1–2.

[7] Yuriy A Reznik, Karl O Lillevold, Abhijith Jagannath, Justin Greer, and Jon Corley. 2018. Optimal design of encoding profiles for abr streaming. In *Proceedings of the 23rd Packet Video Workshop*. 43–47.

[8] Werner Robitza. 2017. CRF Guide. (2017). https://slhck.info/video/2017/02/24/crf-guide.html

[9] SandDrive. 2020. COVID-19 Global Internet Phenomena Report. (2020). https://www.sandvine.com/phenomena

[10] NVIDIA.Disgn works. 2020. GPU-accelerated video processing integrated into the most popular open-source multimedia tools. (2020). https://developer.nvidia.com/ffmpeg

[11] x265.org. 2015. The x265 website. https://x265.org/. (2015).

[12] Deheng Ye, Zhao Liu, Mingfei Sun, Bei Shi, Peilin Zhao, Hao Wu, Hongsheng Yu, Shaojie Yang, Xipeng Wu, Qingwei Guo, et al. 2019. Mastering Complex Control in MOBA Games with Deep Reinforcement Learning. *arXiv preprint arXiv:1912.09729* (2019).

[13] Xiaoqi Yin, Abhishek Jindal, Vyas Sekar, and Bruno Sinopoli. 2015. A control-theoretic approach for dynamic adaptive video streaming over HTTP. In *SIGCOMM 2015*. ACM, 325–338.