# Paper Summary & Critique

| | |
|---|---|
| Paper Title | Improving Mobile Interactive Video QoE via Two-Level Online Cooperative Learning |
| DOI | 10.1109/TMC.2022.3179782 |
| Name | Sahil Pattni |
| Concordia ID | 40216177 |
| Course | COMP 691 |
| Lecture | 7 |

## Summary

The authors of this paper present a novel reinforcement-learning (RL) based adaptive bitrate model - Legato - to improve the quality-of-experience for mobile interactive video . Legato is an online model, which means it can train and adapt to information in real-time. The RL model is paired with the legacy rule-based video bitrate algorithm. An adaptive score mechanism attempts to predict which of the two systems is more likely to produce a higher QoE. The system aggregates the learned intelligence from individual models associated with each user.

## Strengths

- Legato's unique learning system allows for the aggregation of individual learning models, which means that it can adapt to a wide range of network conditions.
- The system works with interactive video, which is generated in real-time and requires extremely low-latency inference to keep up with the video content.
- Legato is able to outperform state-of-the-art solutions.

## Weaknesses

- Widely used machine learning frameworks only allow for RL model inference, not its continuous training on mobile devices. The only framework that the authors could find that supports such a training system is developed in Java, and is not easily compatible with modern interactive video applications.

## Applicability to Practice

As stated by the authors, current platforms that allow for online RL training are niche and not easily compatible with modern interactive video platforms. This poses deployment concerns that need to be addressed before such a system can be distributed at scale.

## Notes for Improvement

While the inference of any given individualized model may not be computationally expensive, the cost of aggregating each model at the server side, as well as re-distributing new aggregate models might be significant, depending on the scale of deployment. In addition to this - and as discussed above - an online RL platform that is more compatible with model video applications may make Legato more accessible to end-users.

| | |
|---|---|
| Paper Title | GRACE: Loss-Resilient Real-Time Video through Neural Codecs |
| DOI | 10.48550/arXiv.2305.12333 |
| Name | Sahil Pattni |
| Concordia ID | 40216177 |
| Course | COMP 691 |
| Lecture | 7 |

## Summary

The authors of this paper present a loss-resilient real-time system - GRACE - which works to maintain a user's QoE across a wide range of packet losses. It does this by simultaneously optimizing the encoder and decoder via a neural video codec.

## Strengths

- GRACE mitigates the size overhead of keeping each packet independently decodable by regularizing the values in its output to conform to the same distribution.
- GRACE is able to maintain the same performance as H.264/H.265 on lossless video and outperforms these codecs at higher loss levels.

## Weaknesses

- GRACE is computationally heavy due to its neural engine, and requires a GPU to run inference. Even with GRACE-lite, the system was only able to achieve around 26 fps [1], which is satisfactory for video-streaming (e.g. Netflix), but not for real-time applications such as cloud gaming and interactive virtual-reality.
- Also due to its dependance on the NVC, GRACE cannot outperform H.264/H.265 codecs at compression efficiency on videos with high spatial complexity.

## Applicability to Practice

If GRACE's computational limitations can be addressed or optimized, it would pave the way for more loss-resilient video codecs which could facilitate better real-time interactive video experiences.

## Notes for Improvement

Whereas GRACE is too computationally heavy, GRACE-lite is not powerful enough to be deployed generally. Admittedly, encoding standard codecs for devices such as an iPhone 14 is also computationally heavy, so perhaps the issue may be resolved as more powerful mobile hardware is introduced (e.g. NPUs).

[1] 26.3 fps encoding, 69.4 fps decoding.