Puffer ABR Analysis

Bryce Wang, Richard Huang, Steven Jiang

Abstract

Explainability tools are increasingly used to interpret complex machine learning models, with growing interest in applying them to systems that make critical, real-time decisions such as Adaptive Bitrate (ABR) streaming algorithms. Prior work has shown that tools like Trustee and Agua can detect shortcut learning and provide interpretable approximations of model behavior. However, it remains unclear whether these tools can accurately recover the full decision-making logic of an algorithm, particularly in structured, rule-based systems. This gap limits our ability to validate whether explanations are truly faithful or just superficial. In this paper, we explore this problem by evaluating Trustee and Agua on their ability to interpret Model Predictive Control (MPC), a deterministic ABR algorithm used in the Puffer streaming platform. We train a classifier to replicate MPC decisions using real-world data, then analyze the outputs of the explainability tools applied to that model. Our findings show that while the tools can surface high-level patterns aligned with MPC's objectives, they fall short in capturing its full semantics, especially aspects involving temporal variation.

1 Introduction

At a high level, this work explores the problem of evaluating the effectiveness of explainability methods in their ability to interpret Adaptive Bitrate (ABR) algorithms. Black-box machine learning models pose challenges to transparency and explainability tools aim to provide insight into their decision-making processes. Trustee and Aqua are examples of such explainability tools. Prior research has shown that they are capable of identifying shortcut learning: when models rely on correlations based on features that aren't pertinent to the subject rather than meaningful semantic features. While this ability is valuable, we seek to push these explainability methods further by evaluating whether they can accurately characterize algorithms. To this end, we focus on ABR algorithms, which deployed in video sharing platforms to dynamically adjust video quality to try to give a good Quality of Experience (QoE). Some ABR algorithms are deterministic and thus serve as an ideal ground truth for comparison to an explainability tool. Our motivation is to assess whether explainability tools like Trustee and Aqua can go beyond detecting shortcuts and reflect the underlying logic of semantically-driven algorithms such as ABR.

To support this goal, we sought a dataset that satisfied two key criteria: (1) a large volume of user session data and (2) an explicit association with a known ABR algorithm. We selected the Puffer [3] dataset, a video streaming study conducted by Stanford, which provides real-world data alongside information about the ABR algorithms used during playback sessions. Puffer includes a mix of ABR algorithms, some of which are based on black-box machine learning models and trying to interpret those ML-based ABR algorithms would pose challenges for interpretability due to their opacity. Instead, we narrow our focus to the subset of sessions using Model Predictive Control (MPC), a deterministic, rule-based ABR algorithm. Because MPC operates based on well-defined heuristics and optimization principles, it serves as an ideal candidate for assessing whether explainability tools like Trustee and Aqua can recover the underlying logic of a semantically grounded algorithm.

In this paper, we show that while existing explainability tools like Trustee and Agua are effective at surfacing high-level patterns in ABR decision-making, they struggle to capture the full semantic logic of algorithms like Model Predictive Control (MPC). Using the Puffer dataset, we train a model to replicate MPC's bitrate selection behavior and evaluate how well Trustee and Agua can recover its core decision principles. We find that Trustee can partially reflect MPC's emphasis on maximizing video quality and minimizing rebuffering, but fails to account for objectives involving quality variation across time. Agua's concept dictionary provides some interpretive gains, but often abstracts away important nuances in decision paths.

At a high level, our work differs from efforts in Trustee [1] in both purpose and methodology. While their results focus on identifying shortcut learning behaviors in models, our goal is more targeted. We aim to see if it is possible to reverse-engineer a specific algorithm used by the model. In addition, we critically examine Agua's claim of enhanced interpretability by testing whether its outputs truly reflect more understandable or transparent reasoning.

The remainder of this paper begins with a detailed background on adaptive bitrate streaming and the specific algorithmic framework used by Puffer, alongside an overview of the interpretability tools Trustee and Agua.

We then describe the design of our modeling pipeline, including dataset preprocessing and classification strategies, as well as key implementation challenges such as class imbalance and shortcut learning. Following this, we present an evaluation of how well Trustee and Agua reflect the underlying logic of the MPC algorithm, with empirical analysis of the strengths and limitations of each tool. We conclude with a discussion of our findings, their broader implications for explainability research, and potential directions for future work.

2 Background and Motivation

2.1 Adaptive Bit Rate (ABR)

Adaptive Bit Rate is a technique commonly used in video streaming designed to optimize the playback experience over variable network conditions. It works by encoding audiovisual content at multiple bitrates and segmenting it into small, time-aligned chunks. During playback, the client continuously monitors network bandwidth, buffer occupancy, and device performance to select the most appropriate bitrate level for each subsequent segment realtime. This adaptive process ensures that playback remains smooth and uninterrupted, even under changing network conditions, by trading off video quality to prevent rebuffering events.

A central challenge in the design and optimization of ABR systems is maximizing Quality of Experience (QoE) for end users. QoE is a holistic metric that captures how users perceive the playback experience, influenced by factors such as video resolution, smoothness of playback, and responsiveness to changes in network conditions. Poor QoE is often associated with frequent rebuffering events, significant drops in video quality, or abrupt quality switches. To tackle this challenge, ABR algorithms try to strike a balance, aiming to provide the highest video quality possible without causing playback to stall. An ideal system smoothly adapts to changing network conditions, always delivering the best quality that the connection can support at the time without interruptions, buffering, or sudden drops in visual clarity.

2.2 Puffer

Puffer is a research platform developed at Stanford University for evaluating adaptive bitrate (ABR) algorithms in real-world video streaming scenarios. Unlike most ABR studies that rely on offline simulations or synthetic network traces, Puffer streams live TV content over the internet to real users and collects detailed performance metrics during playback. This setup allows researchers to test and compare ABR strategies under actual, variable network conditions, where user behavior and traffic dynamics can significantly impact performance.

In addition to serving as a live platform for evaluating ABR algorithms, Puffer provides publicly available datasets that record detailed information about each streaming session. These datasets are collected daily and include logs of every video chunk sent from the server to a client, as well as periodic updates reported by the client. For each chunk, the server log captures important information such as the time it was sent, its encoding format and size, a quality metric called SSIM (structural similarity index), and network conditions including round-trip time (RTT), congestion window size, and delivery rate. On the client side, events such as startup delay, rebuffering, playback resumption, and buffer levels are tracked over time. These logs make it possible to analyze how network performance affects playback quality and how different ABR strategies respond to changing conditions. We leverage these datasets to train and evaluate machine learning models that predict playback quality across various network scenarios.

2.3 Model Predictive Control

Model Predictive Control (MPC) is a general algorithm used to make sequential decisions. In the context of Adaptive Bitrate (ABR) streaming, MPC selects a sequence of future bitrate levels to maximize the user's Quality of Experience (QoE). Puffer's implementation defines this as the

$$QoE = \text{SSIM} - \lambda * \text{SSIM variation} - \mu * \text{stall time} \quad (1)$$

where SSIM is a metric for video quality relative to the canonical source, λ is the weightage on the SSIM variation, and μ is the weightage on the stall time. Essentially, MPC tries to find the best next video quality by maximizing the predicted video quality, minimizing the predicted variation in quality, and minimizing the predicted stall time.

The full objective function optimized by MPC as described in the Puffer paper is:

$$\max_{a_1,\ldots,a_K} \sum_{k=1}^{K} [q(a_k) - \mu\,|q(a_k) - q(a_{k-1})| - \lambda \max(T(a_k) - B_k, 0) \quad (2)$$

where:

- a_k is the bitrate level selected for chunk k,
- $q(a_k)$ denotes the video quality associated with bitrate a_k,
- μ is the penalty weight for bitrate switching (instability),
- $|q(a_k) - q(a_{k-1})|$ penalizes large changes in video quality between consecutive chunks.
- $\max(T(a_k) - B_k, 0)$ describes the stall time experience of sending a_k and is the estimated rebuffering time for chunk k,

- λ is the penalty weight for rebuffering,

Thus, in the broad scope of this project, we hope to see that explainibility models also show that MPC seeks to make bitrate decisions that balance high video quality, minimal rebuffering, and smooth playback transitions.

2.4 Trustee

Trustee [1] is an explainability tool designed to interpret and visualize the decision-making process of machine learning models. It takes in a black-box model as input and trains a decision tree to approximate the predictions of that model. The resulting surrogate tree mimics the behavior of the original model, enabling interpretability. Trustee generates a trust report that outlines how different feature thresholds influence classification decisions. For example, if $x < 10$, the model might favor one class, whereas if $x \geq 10$, it may favor another. This allows users to better understand and validate the logic behind non-transparent models. In the context of this paper, we see if Trustee's decision tree can help reverse engineer the MPC algorithm.

2.5 Agua

The Agua [2] paper builds upon Trustee by introducing the notion of a concept dictionary. The authors argue that Trustee's decision tree can become large, making interpretation overhead demanding. To address this, Agua augments the decision tree with a concept dictionary that maps tree nodes or feature conditions to higher-level, human-understandable concepts. In our evaluation, we empirically assess whether Agua indeed enhances the interpretability of Trustee by examining the effectiveness and usability of the concept dictionary in practice.

3 Design and Implementation

We began with access to Puffer's video segment-level transmission data. However, this dataset lacked information about the playback buffer, which is critical for modeling ABR behavior. To address this, we incorporated additional data containing the buffer window and joined the two datasets using session identifiers and synchronized timestamps. This enriched dataset provided a more complete view of the streaming context necessary for learning ABR decisions. Also, Puffer's qualities are arranged in quality-frame rate format. For our ABR, we ignore the frame rate and simply look at the quality to simplify the problem.

Using this combined data, we trained a Random Forest (RF) classifier to predict the "format" (i.e., video quality level) selected for each video segment. The model initially achieved an F1 score of 0.82, suggesting good performance. However, deeper analysis revealed that the classifier predominantly predicted the majority high-quality formats (1920x1080 and 1280x720) but poorly predicted any lower resolutions with F1's as low as 0.01. This indicates poor generalization due to class imbalance. This turned out to be a very difficult issue to try to resolve. Our methods were

- Randomized class balancing: We would select 25000 random datapoints of each of the 12 labels. This did very bad as the average F1 dropped to below 0.4 since the classifier was unable to accurately predict the majority classes either.
- Weighted RF: The methodology here was to use the imbalanced data we had then change the SKLearn Random Forest parameter "class_weight" to balance the classes more accurately. Specifically we set the weight of each class to $C/MaxSamples$ where C is the number of samples in a class and $MaxSamples$ is the maximum number of samples across all classes. This, at a glance, performed much better than the Randomized class balancing but had the same initial issues as the original Random Forest i.e. the lower quality classes had very accuracy.
- Smotes: To further address class imbalance, we applied SMOTE to generate synthetic samples for the underrepresented lower-quality formats. The expectation was that this would help the classifier generalize better across all classes. However, the results were largely unchanged: the overall F1 score remained around 0.81, and minority class F1 scores stayed low (around 0.4). This suggests that while SMOTE increased the representation of minority classes, it did not introduce enough meaningful variation to improve their predictive performance.
- Non-random class balancing: This follows a similar premise to the randomized class balancing, but instead we would select the first 25000 datapoints per class. This performs much better receiving an average F1 of 0.76 overall with the major setbacks still coming from the lower classes but instead of their F1's being 0.01, the lowest F1 was 0.23. We hypothesize that since networking data is continous especially in the context of ABR, it doesn't make sense to split the data randomly based off the classes. As in, ABR is based off the continuous stream of networking data to adjust based off the network conditions, so giving a model data thats scattered doesn't make sense.

We ended up following through with non-random class balancing as it obtained the highest F1 and highest lower bound on all other classes. We do acknowledge that the F1 could have been better but yeah we didn't know of any other options.

When deploying the model in Trustee and Aqua, we also encountered an instance of shortcut learning. Upon investigation, we realized that session identifiers had not

been removed from the feature set, allowing the model to trivially memorize per-session behaviors rather than generalizing from network and buffer conditions. This was the same style of shortcut learning presented by Trustee [1] in their results. After removing this leakage, we re-evaluated the model under more semantically relevant constraints and obtained the F1 scores shown above.

These simulation runs help us evaluate how closely our inferred model replicates Puffer's decision logic and how it responds to varying network conditions. The insights gained will inform further refinement of both the model and our understanding of the underlying ABR strategies. In the following section we describe an analysis on Trustee and Aqua for explaining ABR conditions.

4 Evaluation

4.1 Trustee

Our initial expectation for the output of Trustee was that its decision tree would capture the three key objectives of the MPC equation (Equation 1): (1) maximize video quality, (2) minimize video quality variation across chunks, and (3) minimize rebuffering. While the first and third objectives will be discussed later in this section, we observed that Trustee is not well-suited to capture the second objective. Specifically, Trustee operates at the level of individual data points, rather than over sequences or neighborhoods of points. Consequently, it lacks the semantic capacity to reason about variation in quality across multiple video chunks. This limitation makes it inherently difficult for Trustee to express or explain behaviors related to inter-chunk variability, which is a core component of the second MPC objective.

In terms of the first and third objectives, we present Figure 1. In analyzing the Trustee-pruned decision tree, we found that its top-level splits included features such as buffer level, chunk size, delivery rate, and cumulative rebuffering, variables closely tied to the core objectives of maximizing quality and minimizing rebuffering. While the model does not explicitly capture the objective of minimizing quality variation, this is consistent with its chunk decision structure, which lacks temporal awareness. However, despite being a decision tree, the interpretability of Trustee was not as clear as expected. The resulting tree structure, though pruned, often produced decision paths that were nontrivial to trace or explain, making it difficult to extract meaningful, high-level insights. Overall, the Trustee model appears to learn heuristics aligned with MPC behavior, such as selecting smaller chunks under low buffer or poor delivery conditions, although its lack of sequence modeling limits its ability to fully reflect all MPC tradeoffs.

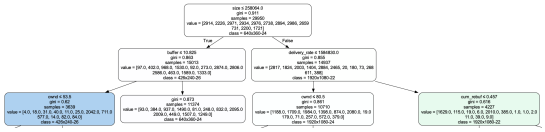

Figure 1: *Trustee Decision Tree for Random Forest Classifier.*

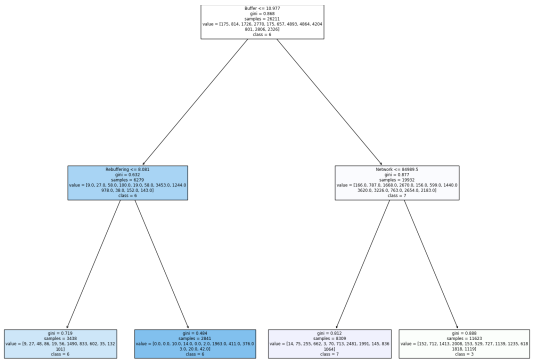

Figure 2: *Agua interpretability with concept dictionary on Puffer MPC Random Forest model.*

4.2 Agua

In this section, we empirically evaluate Agua's claim that incorporating a concept-based dictionary enhances model interpretability. As shown in Figure 2, the level of interpretability provided by Agua appears to be comparable to that of the original Trustee model. While the concept dictionary introduced some minor changes in phrasing, these modifications did not significantly improve the clarity or usefulness of the explanations. Based on our evaluation, the dictionary provided by Agua does not substantially enhance interpretability in practice.

5 Conclusion

Our analysis demonstrates that explainability tools like Trustee and Agua can recover some, but not all, of the underlying logic behind deterministic ABR algorithms such as Model Predictive Control (MPC). While they are capable of capturing high-level objectives like maximizing video quality or avoiding rebuffering, they struggle with more nuanced goals such as minimizing quality variation over time. These findings highlight a gap between surface-level interpretability and deeper semantic understanding, especially in systems that depend on temporal reasoning.

5.1 Limitations

While our study provides insight into the capabilities and limitations of explainability tools for ABR algorithms, it is not without limitations. First, although MPC is a deterministic and rule-based algorithm, our analysis relies on a trained classifier to approximate its behavior. This introduces modeling noise, and any discrepancy between the classifier's outputs and the true MPC logic weakens

the validity of our conclusions about Trustee and Agua's ability to reverse engineer the algorithm. Second, our focus on a just MPC limits the generalizability of our findings. While its transparent, rule-driven nature made it a good baseline for interpretability analysis, the behavior of explainability tools may differ significantly when applied to more complex or opaque models, such as those based on deep learning. Future work should explore a broader range of algorithms to better understand the applicability of these tools across different algorithms.

5.2 Future Work

Our study lays the groundwork for evaluating explainability tools in the context of ABR decision-making, but several directions remain open. One natural extension is to apply our analysis pipeline to machine learning–based ABR algorithms such as Pensieve or RobustMPC. These models present a greater interpretability challenge due to their opaque, black-box nature, and would allow us to assess how Trustee and Agua perform when the underlying logic is unknown or less structured than in MPC. Additionally, to move beyond qualitative assessments, future work should explore the development or adoption of formal interpretability metrics. These would enable more rigorous, quantitative comparisons between explainability tools by measuring how closely their outputs align with the original algorithm.

References

[1] JACOBS, A. S., BELTIUKOV, R., WILLINGER, W., FERREIRA, R. A., GUPTA, A., AND GRANVILLE, L. Z. AI/ML for Network Security: The Emperor has no Clothes. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security (CCS '22)* (2022), ACM. (Cited on pages 1, 3 and 4.)

[2] PATEL, S., HAN, D., NARODYTSKA, N., AND ABDU JYOTHI, S. Toward Trustworthy Learning-Enabled Systems with Concept-Based Explanations. In *Proceedings of the 23rd ACM Workshop on Hot Topics in Networks* (2024), ACM. (Cited on page 3.)

[3] YAN, F. Y., AYERS, H., ZHU, C., FOULADI, S., HONG, J., ZHANG, K., LEVIS, P., AND WINSTEIN, K. Learning in situ: a randomized experiment in video streaming. In *Proceedings of the 17th USENIX Symposium on Networked Systems Design and Implementation (NSDI 20)* (2020), USENIX Association. (Cited on page 1.)