Pitfalls of data-driven networking: A case study of latent causal confounders in video streaming

P. C. Sruthi
psruthi@purdue.edu
Purdue University

Sanjay Rao
sanjay@purdue.edu
Purdue University

Bruno Ribeiro
ribeiro@cs.purdue.edu
Purdue University

ABSTRACT

This paper motivates the need to support counterfactual reasoning (i.e., answer "what-if" questions about events that did not occur) when collecting network data. We focus on video streaming – e.g., given logs of a video session, a video publisher may ask whether a user would continue to experience no rebuffering events if the lowest quality video choice were eliminated. We discuss potential pitfalls related to counterfactual reasoning, and argue that dynamic network state (e.g., bandwidth) serves as a confounding yet hidden (latent) feature that complicates such analyses. We illustrate the challenges, and present preliminary methods to address them using concrete examples. Our evaluations show that existing approaches, including randomized trials (collecting data from an algorithm that selects bitrates randomly), are by themselves inadequate for counterfactual reasoning related to video streaming, and must be supplemented by techniques that explicitly infer latent features.

CCS CONCEPTS

• **Networks** → **Network performance modeling**; • **Information systems** → **Multimedia streaming**.

KEYWORDS

Data-driven networking, Video delivery, Adaptive bitrate algorithms

ACM Reference Format:
P. C. Sruthi, Sanjay Rao, and Bruno Ribeiro. 2020. Pitfalls of data-driven networking: A case study of latent causal confounders in video streaming. In *Workshop on Network Meets AI & ML (NetAI'20), August 14, 2020, Virtual Event, NY, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3405671.3405815

1 INTRODUCTION

When designing networked systems in a manner informed by data from real deployments, it is common to ask "what-if questions", on the potential impact if an alternate choice had been made. For instance, given data collected from real video streaming sessions, a video publisher may wish to understand the performance if a different Adaptive Bitrate (ABR) algorithm were used. Alternately, a publisher may wish to understand whether a user would continue to experience no rebuffering events if the lowest quality video choice were eliminated. In analyzing such questions, it is useful to ask what the download time of a video segment could have been if a bitrate different than the one selected in the actual session had been chosen.

Answering what-if questions of this nature is also known as *counterfactual reasoning*. Counterfactual reasoning considers the effect of events that did not occur while the data was being recorded [13], and is often used in fields such as epidemiology [14]. Unfortunately, such reasoning with networking data is challenging because many networked systems are adaptive in that they make decisions based on the network state (e.g., bandwidth, latency), and this state itself impacts any observable measurements (e.g., download times). For instance, Internet video is typically split into chunks, with each chunk encoded at multiple bitrates. ABR algorithms decide what bitrate to pick for each chunk based on their perception of network bandwidth, but the bandwidth in turn determines the download time of the chunk. Thus, the network bandwidth acts as a *confounding variable* that makes it difficult to easily infer the download time that a video chunk would see if an alternate decision had been made by the algorithm.

While some prior work considers what-if questions (e.g., [17]), they do not consider the biases arising out of confounding variables. Other works have recognized the need to account for confounding variables when analyzing network data [3, 10, 18], though they do not consider counterfactual reasoning. Further, these works only deal with *observable* confounding variables such as a user's connection type (i.e., whether a user is behind a DSL, cable modem or WiFi), for which information is available as part of the data. Unfortunately, variables that capture dynamic network state (e.g., bandwidth, latency) are *latent*, and require new methodologies not covered in prior work.

In medical studies, causal effects are directly measured by randomized controlled trials (RCTs)[4], eliminating the need to deal with latent confounders. In networking, for instance, this would imply that rather than using an ABR algorithm that picks video bitrates based on perceived network state, we could collect data from video sessions where chunk sizes are picked at random. Unfortunately, we show that RCTs are not a silver bullet in networking applications: we still need to deal with latent confounders, otherwise, in the example above, RCTs would have no effect on our ability to measure the counterfactual effect of an ABR algorithm picking different bitrates.

Motivated by these observations, we argue for a research methodology that explicitly accounts for such latent confounding variables. Our approach involves (i) identifying the causal dependency graph aided by domain knowledge; (ii) identifying measured and latent confounders; (iii) inferring latent features; and (iv) grouping data

NetAI'20, August 14, 2020, Virtual Event, NY, USA

ACM ISBN 978-1-4503-8043-0/20/08…$15.00
https://doi.org/10.1145/3405671.3405815

on the inferred latent features. We illustrate the approach for a case study with video streaming, and an example bandwidth process. The step of inferring latent features is a crucial step which we achieve using Maximum A Posteriori estimation in our case study. While the bandwidth process we consider may seem simplistic, it is nevertheless useful in establishing the importance of accounting for latent confounders.

We present empirical results that show that latent confounding variables must be accounted for while evaluating what-if questions in these systems, and that —because of these latent confounders— widely used approaches like direct emulation from data incorrectly estimate the effects of changes in video streaming systems. Our results also show that accounting for latent confounders is critical to ensure good results with randomized trials and observational studies.

2 CAUSALITY IN VIDEO STREAMING

2.1 The need for causal inference

In video streaming, a video is split into chunks, each encoded at multiple bitrates. To cope with variability in network bandwidth, clients run Adaptive BitRate (ABR) algorithms [2, 6, 12, 16, 19, 20]. For each chunk, these algorithms select what bitrate to pick based on the client's estimated throughput, and the amount of data stored in the client buffer.

Consider records of a video session with N chunks. These records minimally contain a set of tuples of the form Data $= \{(t_i, s_i, d_i)\}_{i=1}^{N}$, where

t_i : start time of i-th video chunk download request,
s_i : the i-th video chunk size,
d_i : the i-th chunk's download time.

Using this data, we wish to answer the following what-if counterfactual question: *for a given $i \in \{1, \ldots, N\}$, what would have been the download time d_i', if chunk i had been downloaded starting at time t_i, and if we had requested chunk size $s_i' \neq s_i$?* In Section 3, we show that under some conditions, even assuming no latency or transport layer effects, one cannot accurately estimate d_i' using d_i, s_i, s_i', and t_i. Without an accurate estimate of d_i', we are unable to precisely evaluate the potential of an alternate ABR algorithm.

Here, the challenge of counterfactual evaluation comes from the causal relationships among observed and latent (hidden) variables, some of which may be *confounders*. Consider three variables X, Y and Z. If X influences Y, but Z influences both X and Y, Z is a confounder, because it introduces a correlation/dependence between X and Y even in the absence of a direct connection between X and Y. Confounders are easier to handle if their values are directly observed as part of the data (e.g., whether a user is connected behind cable or wireless [10]). However, we illustrate below in the context of video streaming, that confounders may be *latent*. Consequently many existing techniques do not directly apply as we discuss in Section 2.2.

Example A: Figure 1 illustrates one of the simplest causal dependencies in a video streaming session employing an ABR algorithm. The chunk size S and download time D are both observed in the data but depend on the *latent variable* B (the true network bandwidth, which is not observed), since the ABR algorithm chooses a video

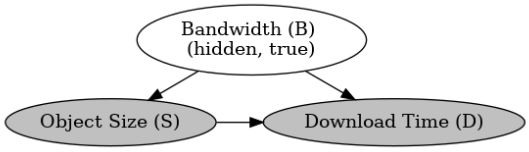

Figure 1: Causal relationships between size, bandwidth, and download time. The true bandwidth (unshaded) is hidden, while the others are measured and available in the data.

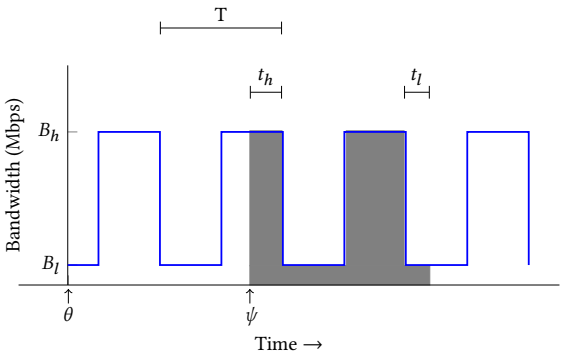

Figure 2: An example time-varying bandwidth process

chunk size S based on a prediction of the value of the bandwidth B, which therefore acts as a confounding variable for both S and D.

To understand the effect of latent confounder B on counterfactual reasoning in ABR algorithms, consider the following scenario. Assume $B \in \{1, 2\}$ MBps and whenever $B = 1$ we have the ABR algorithm choosing $S = 1$ MB which implies $D = 1$ second; and when $B = 2$ we have the ABR algorithm choosing $S = 2$ MB which again implies $D = 1$ second. A pure data-driven approach would conclude that $P(D = 1|S = 1) = P(D = 1|S = 2) = 1$. Hence, statistically, it seems like $D = 1$ regardless of the choice of S. The above mistake comes from confusing $P(D|S)$, which we directly obtain from the data, with the counterfactual query $P(D|do(S))$, which is the distribution of download times if the size had been *forced* to be S.

In the above example, it may appear that one could estimate $\hat{B} = S/D$ (i.e., the average throughput observed during the download), and use this information in the conditional. But, the true bandwidth B may vary *during a download*, resulting in the estimation $\hat{B}$ not being theoretically sound as we illustrate next.

Example B: Now consider a time-varying bandwidth process $B(t)$ in the square wave pattern in Figure 2. The peak and trough values are B_h and B_l respectively with the period of the wave (starting and ending at a falling edge) being T (these are assumed known). The shaded region corresponds to when a particular chunk is downloaded. Clearly, chunks of different sizes may see different average observed throughput even if their downloads were to start at the same time. Further, for a given size, the download time depends on the phase ψ of the square wave at which the chunk starts downloading (henceforth referred to as the chunk start phase).

The causal graph in this setting changes from the one in Figure 1 to the one in Figure 3 as we will elaborate in Section 3.1. For

any chunk, the start phase ψ is a confounding variable since (i) it determines the instantaneous bandwidth perceived by the ABR algorithm and consequently which chunk size is selected; and (ii) it impacts the download time for a chunk of a given size. Further, ψ is latent since information about ψ is not directly available as part of the data.

Also note that the size of the first chunk affects the start time of the second chunk. This in turn affects the start phase and download time of the second chunk, which in similar fashion affects the start time, start phase and download time of subsequent chunks. More generally, $B(t)$ induces a correlation between both the start phase and size of the j-th chunk, and the download time of the i-th chunk, $j \leq i$. Because of these complex dependencies, simply estimating the bandwidth as $\hat{B} = S/D$ is not enough to predict how chunk sizes affect download times (we also show this experimentally later). Further, this also impacts the effectiveness of standard approaches to deal with biases such as randomized trials, as we discuss below.

A potential approach to aid counterfactual reasoning is to collect download time data based on fixed-size data blocks at a finer granularity. While this may provide more observations regarding the bandwidth process, it does not eliminate the challenges in dealing with latent variables. This is because, owing to the on/off patterns of video streaming transmissions [1, 7], there may be periods for which no data on available bandwidth is available. Further, it may not always be possible to collect bandwidth information with sufficiently fine granularity.

2.2 Existing approaches and related work

We next discuss the primary methods used for causal reasoning and the challenges in using them:

• **Controlled experiments or Randomized Controlled Trials (RCTs).** RCTs [4] are widely considered to be the gold standard for measuring the effects of interventions. An RCT for bitrate selection would choose chunk sizes randomly. This approach forces randomness by picking video bit rates in a manner that does not consider bandwidth for some sessions. Even if RCTs in real-world video streaming sessions were viable, their usefulness for causal inference is somewhat limited in the video streaming context. This is because a chunk's download start time still depends on the download end time of the previous chunk. Hence, the effect of the previous and current chunk sizes on the download times are not independent. In Section 4, we present evaluations to show that simply using RCTs without accounting for the latent variables is insufficient for accurate inference.

• **Observational studies.** RCTs might not always be viable in practice – e.g., in our context, they could result in degraded quality to users in real-world video streaming sessions. An alternate approach involves using an offline analysis with collected data, but accounting for *observed* confounding factors during the analysis – i.e., the confounding variable values are available as part of the data, such as user connection type (DSL vs. mobile), and geography (country). The approach has been applied in networking [9, 10, 18] – e.g., Krishnan et al. [10] consider how user connection type and geography confound correlations between video performance and user engagement. None of these works consider counterfactual reasoning. Further, while controlling on measured confounders removes some of the bias resulting from the data collection process, dealing with latent confounders (such as dynamic network state) is more challenging, and plays an important role, as we show empirically in Section 4.

The work of Bartulovic et al. [3] seeks to correct the bias that smaller chunk sizes may see poorer throughput than larger ones owing to TCP slow start effects — unlike our work, it does not model the causal relationship that the ABR algorithm itself may pick chunk sizes in a manner that depends on its perception of bandwidth. To this end, when comparing the total reward seen by algorithm B on a trace collected from an algorithm A, Bartulovic et al. only consider those video chunks where the new algorithm picks the same bitrate as the old algorithm. This evaluation process is viable as Bartulovic et al. only consider a *constant bandwidth process* and their importance reweighting approach simply involves comparing bitrate choices of the two algorithms, the data for which is directly available. Unfortunately in our context, the bandwidth process is *latent* and not directly available as part of the data. Moreover, we are interested in counterfactual queries: "What would have been the download time of the chunk if algorithm A had chosen chunk size s' rather than s?".

• **What-if analysis and causal graph inference.** Another set of works [8, 15, 17] consider *what if* analyses for various applications, but do not consider how to perform such analyses in the presence of confounding or latent confounding variables. More closely related works [5, 9, 11, 17] have looked at inferring the dependency graph of measured features directly from the data, and some [9, 11] do account for *observable* confounding features. Unfortunately, latent confounders are not considered, and in general, causal graphs inferred from passive data alone may be inaccurate [3, 11].

3 SOLUTIONS: COUNTERFACTUAL REASONING WITH LATENT FEATURES

We now discuss a potential approach to enabling counterfactual reasoning in the presence of latent confounders such as dynamic network state (e.g., bandwidth), a challenge unaddressed by prior work (Section 2.2). We introduce the necessary steps to achieve this goal, and apply it to a concrete case study in the context of video streaming. The key steps are below:

(1) **Identify the causal dependency graph.** This involves using domain knowledge to determine the features that affect the quantity that we are trying to predict using an appropriate model of network state such as bandwidth.
(2) **Identify measured, and latent confounders.** We identify from the causal graph, both confounding features that are *observable* (with information available in the data), and *latent* (information not directly available).
(3) **Inferring latent features.** This is the key new step in the methodology, and the most challenging, where we seek to infer any latent features that are not directly measured.
(4) **Matching and backdoor adjustment.** Finally, we perform a backdoor adjustment over the latent confounding variables and condition over the observable confounders. This way, we block the backdoor over the observable and latent confounders, and can estimate the probability distribution of download times given an intervention on the chunk size.

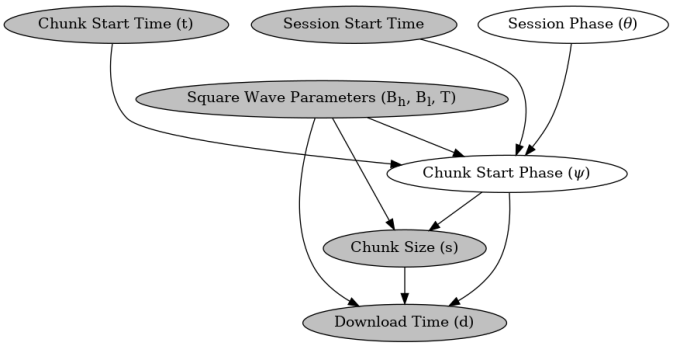

Figure 3: Causal graph for our case study. The shaded ovals represent features that are observable, while the other ovals correspond to latent features.

This procedure is challenging in general, but we outline how it may be achieved in an initial case study, deferring more general contexts to future work.

3.1 Case Study

Consider again the bandwidth model corresponding to Figure 2. Consider that the peak (B_h), trough (B_l), and the time period (T) are known. However, the phase of the wave at the start of the session (henceforth referred to as the session phase θ) is not known. Further, objects of known sizes are downloaded sequentially starting at time $t = 0$. We apply the general procedure described above to this context to estimate the download time of an object if its size had changed.

Identify the causal graph. Figure 3 shows the causal graph for the square wave model, for a specific object download. If the ABR algorithm chooses bitrates based on the instantaneous bandwidth measurement as described in section 2, the chunk start phase ψ determines the instantaneous bandwidth, and hence the chosen bitrate. Further, ψ and the parameters of the square wave, along with the chunk size determine the download time of the chunk.

Identify measured and latent confounders. In the causal graph (Figure 3), we do not know the session phase θ, nor the current chunk's phase ψ. Out of these the chunk's start phase is required to determine the download time, and the session phase is required to determine the chunk start phase. Here, the chunk start phase is a confounder - it affects the chunk size, but it also determines the download time. Further, this is a latent confounder, since the start phase is not provided as part of the data. The chunk start phase plays a similar role to *Bandwidth (B)* in Figure 1.

Infer latent confounders. The chunk start phase is not directly measured by the video player, but is necessary for estimating the download time. In this case, it is possible to infer an estimate for the chunk start phase using the data. Let X_j be the set of all observable variables at the time chunk $j+1$ is about to be requested. We do this by estimating the posterior over the session start phase θ, given the chunks that have been observed:

$$P(\theta|X_j) \propto P(X_j|\theta)P(\theta).$$

We assume a uniform prior distribution for θ, so $P(\theta) = 1/T$. We compute $P(X_j|\theta)$ for each chunk by splitting its download time into

Condition on t_i^h and t_i^l	Conditions on the chunk start phase (ψ_i) (at time ts_i, the start time of the chunk)
$0 < t_i^l < \frac{T}{2}, 0 < t_i^h < \frac{T}{2}$	$\psi_{1,i}^{\text{pred}} = \frac{T}{2} - t_i^l, \psi_{2,i}^{\text{pred}} = T - t_i^h$
$t_i^l = 0, 0 < t_i^h < \frac{T}{2}$	$\psi_i^{\text{left}} = \frac{T}{2}, \psi_i^{\text{right}} = T - t_i^h$
$t_i^h = 0, 0 < t_i^l < \frac{T}{2}$	$\psi_i^{\text{left}} = 0, \psi_i^{\text{right}} = \frac{T}{2} - t_i^l$
$t_i^l = \frac{T}{2}, 0 < t_i^h < \frac{T}{2}$	$\psi_i^{\text{left}} = T - t_i^h, \psi_i^{\text{right}} = T$
$t_i^h = \frac{T}{2}, 0 < t_i^l < \frac{T}{2}$	$\psi_i^{\text{left}} = \frac{T}{2} - t_i^l, \psi_i^{\text{right}} = \frac{T}{2}$
$t_i^l = 0, t_i^h = 0$	$\psi_{1,i}^{\text{pred}} = \psi_{2,i}^{\text{pred}} = 0$
$t_i^l = 0, t_i^h = \frac{T}{2}$; $t_i^h = 0, t_i^l = \frac{T}{2}$	$\psi_{1,i}^{\text{pred}} = 0, \psi_{2,i}^{\text{pred}} = \frac{T}{2}$

Table 1: Conditions on t_i^h and t_i^l to calculate phase

three segments : (1) from the download start time to the start of the next trough of the square wave, (2) the longest possible sequence of full periods of the square wave contained within the download time, and (3) the remainder. The time spent in segments (1) and (3) combined can be further subdivided as t^h (time spent at the peak bandwidth) and t^l (time spent at the trough bandwidth). t_i^h and t_i^l are the amounts of time chunk i spends being downloaded in the high and low sections of the square wave, outside of the complete time periods. They are given by:

$$t_i^h = \frac{(d_i \mod T)B_l - (s_i \mod (T(\frac{B_h+B_l}{2})))}{B_l - B_h}$$

$$t_i^l = \frac{(d_i \mod T)B_h - (s_i \mod (T(\frac{B_h+B_l}{2})))}{B_h - B_l}$$

Each condition on t_i^h and t_i^l translates to two possible distributions on the chunk start phase: (1) two equally likely point estimates for the chunk phase, ψ_1^{pred} and ψ_2^{pred} (2) a uniform distribution over the interval specified by bounds ψ^{left} and ψ^{right}. The conditions for ψ_i are given in Table 1.

Since the square wave is periodic in T, we can infer a distribution estimate for the session start phase θ from each chunk's start phase by performing a time shift to $t = 0$ from the chunk start time ts_i:

$$\theta_{j,i}^{\text{pred}} = (\psi_{j,i}^{\text{pred}} - (ts_i \mod T)) \mod T$$

$$\theta_i^{\text{left}} = (\psi_i^{\text{left}} - (ts_i \mod T)) \mod T$$

$$\theta_i^{\text{right}} = (\psi_i^{\text{right}} - (ts_i \mod T)) \mod T$$

The possible session phases are then the set of θ that are consistent with the distribution estimates for all chunks, and is given by:

$$P(X_j|\theta) = \mathbf{1}\left(\theta \in (\max(\theta_i^{\text{left}}), \min(\theta_i^{\text{right}}))\right) \times \left(\mathbf{1}\left(\theta \in \bigcap_{i=1}^{j}\{\theta_{1,i}^{\text{pred}}, \theta_{2,i}^{\text{pred}}\}\right) + \mathbf{1}\left(\bigcap_{i=1}^{j}\{\theta_{1,i}^{\text{pred}}, \theta_{2,i}^{\text{pred}}\} = \phi\right)\right),$$

where $\mathbf{1}(\cdot)$ is an indicator function. This implies that θ can either lie in a continuous range where all θ's within that range are equally likely (unlikely to happen, since we would have enough points to narrow it down), or that there are a set of disjoint points that are all equally likely.

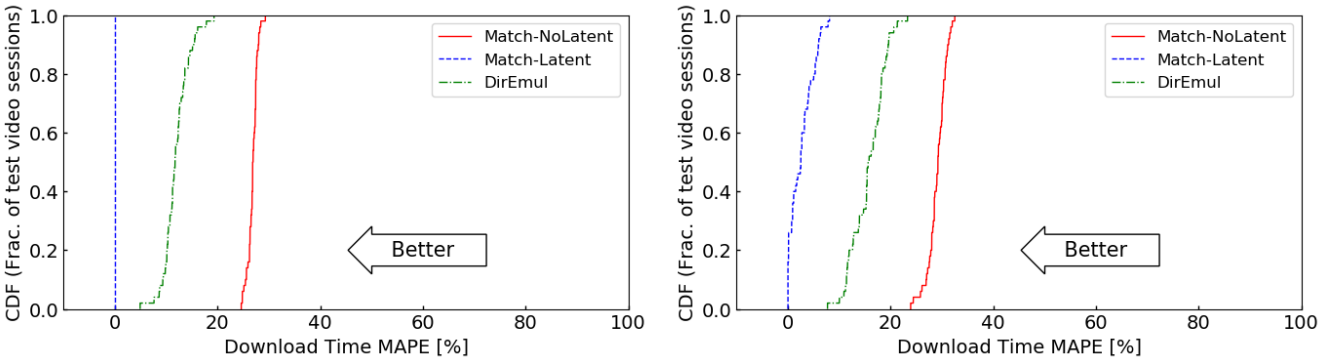

Figure 4: CDFs of Mean Absolute Prediction Error (MAPE) of download time across video sessions for different prediction techniques under two different training data production methods **(a)** RCT - Randomized Controlled Trials **(b)** ABR - Observational Study (see text)

At this point we have an estimate for $P(\theta|X_j)$, which we can use to calculate the chunk start phase ψ by sampling from $P(\theta|X_j)$ and performing a time shift to each chunk's start time to obtain $P(\psi_j|X_j)$.

Matching and backdoor adjustment. Once we have a posterior probability distribution over the start phase of each chunk, $P(\psi_j|X_j)$, we can use the backdoor adjustment [13, Theorem 3.3.2] to estimate the causal effect of changing chunk sizes on the download time, following the causal graph in Figure 3. This entails marginalizing

$$P(d_{j+1}|\text{do}(s_{j+1}), X_j) = \int_{\psi_j} P(d_{j+1}|s_{j+1}, \psi_j, X_j) P(\psi_j|X_j) d\psi_j.$$

Figure 3 shows that ψ_j fully determines the download time (since we know B_h, B_l, T). We take the mean download time of all the chunks in the original data that match on a given input chunk start phase and size, and use that as our estimate for the download time of that chunk.

4 EVALUATION

In this section, we evaluate the importance of explicitly accounting for latent confounding variables when supporting counterfactual reasoning using a simulated video player. We create a training set by simulating the download of a single 5000 chunk video, with chunks encoded at two different bit rates: 1 Mbps, and 2 Mbps. The underlying bandwidth process is the square wave in Figure 2 with high and low bandwidth parameters B_h = 2 Mbps, and B_l = 1 Mbps, following the same notation as in Section 3. The time period (T) of the wave is 5s, and the session start phase (θ) is chosen randomly.

Training dataset production. Our evaluations are conducted in the context of two potential approaches for dealing with causal reasoning, which are defined by the way the training dataset is produced: (i) ***Randomized Controlled Trials (RCT).*** Here, we produce a training trace data set by selecting a random bitrate for each chunk during the download process. (ii) ***Observational studies of ABR algorithms (ABR).*** We produce a training data set by considering a simplified yet illustrative ABR algorithm that selects bit rates based on network throughput. Specifically, we assume the ABR algorithm (1) measures the instantaneous network bandwidth at the start of a chunk download; (2) selects a chunk at the highest bitrate below the instantaneous bandwidth 90% of the time; and (3) selects a chunk at a different bitrate the rest of the time (corresponding to erroneous decisions).

Methodology. Under both data production methods (RCT and ABR), we evaluate the following techniques for estimating chunk download times: (i) ***Direct Emulation (DirEmul).*** Here we use the observed throughput trace as a model for the bandwidth emulating a video download on the trace. (ii) ***Match-NoLatent***. We match on the available observed variables that influence download time i.e. bitrate. The download time estimate for a test chunk is the average download time of all matching chunks in the training dataset. This is similar to the methodology used in Bartulovic et al. [3], wherein matching was done on bitrate for an underlying constant bandwidth model. (iii) ***Match-Latent***. We proceed as in (ii) but filter the training trace observations according to the the bitrate as well as the latent *chunk phase* feature. We infer the session start phase, and calculate the start phases for each chunk as described in Section 3.

RCT and ABR are used to simulate the download of a 5000 chunk (of duration 1s each) video, and the various features for each chunk (bitrate, chunk size, download start time, download time, etc.) are recorded for the training data. For testing, we simulate the download of 50 test videos of 50 chunks each, whose bitrates are chosen randomly as 1 Mbps or 2 Mbps, and predict their download times with DirEmul, Match-NoLatent and Match-Latent. We also obtain the ground truth download times by simulating the test video download on the true bandwidth process. We use the error metric Mean Absolute Prediction Error (MAPE) of chunk download time, calculated for each video by computing the absolute prediction error for each chunk, and taking an average across chunks.

Performance with Randomized Controlled Trials. The results of this simulation are shown in Figure 4(a). We see that when the bitrate selection algorithm that produces the training trace is not adaptive (i.e. it chooses the chunk bitrates randomly), Match-Latent produces an optimal result. Match-NoLatent does poorly, and while DirEmul does slightly better, the error can still be bad

($\approx$ 20%) in the worst case. Even though RCTs are considered to be the de facto standard for estimation of causal effects, we see here that RCT alone is not enough for video streaming; it is critical to also perform a backdoor adjustment on the latent confounder (the chunk phase).

Performance with Observational Studies of an ABR algorithm. In Figure 4(b) we see that when the training data is collected with our simpified ABR algorithm, Match-Latent provides an almost 10 percentage point improvement at the median over DirEmul, and a greater than 15 percentage point improvement over Match-NoLatent. Similar to the RCT case, we see that the inclusion of latent variables in the backdoor adjustment provides a clear improvement over matching on observable features alone. Recall from Section 2.2 that prior work in networking [3, 9, 10, 18] that corrects for biases only considers matching on observable features, and does not consider latent features.

5 CONCLUSIONS AND FUTURE WORK

In this paper, our primary contribution is to draw attention to the challenges posed by latent confounding variables when enabling counterfactual reasoning with network data. We have illustrated the challenges and sketched a potential solution direction in the context of video streaming. Our evaluations show the importance of accounting for latent variables in both Randomized Controlled Trials and Observational Studies, while also showing the limitations of direct emulation methods.

Our work is preliminary in many ways. In the case study considered in this paper, we have assumed that much about the bandwidth process $B(t)$, apart from the session and chunk start phase, is known. In general, the bandwidth process is unknown, which makes our task more challenging. Further, the underlying network bandwidth can be affected by how much data is transmitted (e.g., owing to ISP rate-limiting policies). Our examples show that knowing how to model the bandwidth process is key in being able to estimate counterfactual effects using the session data of an ABR algorithm. Essentially, for correctly performing the backdoor adjustment (Section 3), we need to find an accurate posterior distribution over the bandwidth process into the future given the past observations. Since determining a precise posterior distribution over the network bandwidth process is an open problem, a general solution is outside the scope of this work, and part of our ongoing investigations.

6 ACKNOWLEDGEMENTS

This work was funded in part by the National Science Foundation (NSF) Awards ICE-T:RC 1836889, CCF-1918483, CAREER IIS-1943364, the U.S. Army Research Laboratory contract number W911NF-09-2-0053, and the Purdue Integrative Data Science Initiative. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of NSF.

REFERENCES

[1] Saamer Akhshabi, Lakshmi Anantakrishnan, Ali C Begen, and Constantine Dovrolis. 2012. What happens when HTTP adaptive streaming players compete for bandwidth?. In *Proceedings of the 22nd International Workshop on Network and Operating System Support for Digital Audio and Video*. ACM, 9–14.

[2] Zahaib Akhtar, Yun Seong Nam, Ramesh Govindan, Sanjay Rao, Jessica Chen, Ethan Katz-Bassett, Bruno Ribeiro, Jibin Zhan, and Hui Zhang. 2018. Oboe: auto-tuning video ABR algorithms to network conditions. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication - SIGCOMM '18*. ACM Press, Budapest, Hungary, 44–58. https://doi.org/10.1145/3230543.3230558

[3] Mihovil Bartulovic, Junchen Jiang, Sivaraman Balakrishnan, Vyas Sekar, and Bruno Sinopoli. 2017. Biases in Data-Driven Networking, and What to Do About Them. In *Proceedings of the 16th ACM Workshop on Hot Topics in Networks - HotNets-XVI*. ACM Press, Palo Alto, CA, USA, 192–198. https://doi.org/10.1145/3152434.3152448

[4] Angus Deaton and Nancy Cartwright. 2018. Understanding and misunderstanding randomized controlled trials. *Social Science & Medicine* 210 (Aug. 2018), 2–21. https://doi.org/10.1016/j.socscimed.2017.12.005

[5] Hadrien Hours, Ernst Biersack, and Patrick Loiseau. 2015. A Causal Approach to the Study of TCP Performance. *ACM Transactions on Intelligent Systems and Technology* 7, 2 (Dec. 2015), 25:1–25:25. https://doi.org/10.1145/2770878

[6] Te-Yuan Huang, Ramesh Johari, Nick McKeown, Matthew Trunnell, and Mark Watson. 2014. A buffer-based approach to rate adaptation: evidence from a large video streaming service. In *Proceedings of the 2014 ACM conference on SIGCOMM - SIGCOMM '14*. ACM Press, Chicago, Illinois, USA, 187–198. https://doi.org/10.1145/2619239.2626296

[7] Junchen Jiang, Vyas Sekar, and Hui Zhang. 2012. Improving Fairness, Efficiency, and Stability in HTTP-based Adaptive Video Streaming with FESTIVE. In *Proceedings of the 8th International Conference on Emerging Networking Experiments and Technologies (CoNEXT '12)*. ACM, 97–108. https://doi.org/10.1145/2413176.2413189

[8] Yurong Jiang, Lenin Ravindranath Sivalingam, Suman Nath, and Ramesh Govindan. 2016. WebPerf: Evaluating What-If Scenarios for Cloud-hosted Web Applications. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication - SIGCOMM '16*. ACM Press, Florianopolis, Brazil, 258–271. https://doi.org/10.1145/2934872.2934882

[9] Satoru Kobayashi, Kazuki Otomo, Kensuke Fukuda, and Hiroshi Esaki. 2018. Mining Causality of Network Events in Log Data. *IEEE Transactions on Network and Service Management* 15, 1 (March 2018), 53–67. https://doi.org/10.1109/TNSM.2017.2778096 Conference Name: IEEE Transactions on Network and Service Management.

[10] S. Shunmuga Krishnan and Ramesh K. Sitaraman. 2012. Video stream quality impacts viewer behavior: inferring causality using quasi-experimental designs. In *Proceedings of the 2012 Internet Measurement Conference (IMC '12)*. Association for Computing Machinery, Boston, Massachusetts, USA, 211–224. https://doi.org/10.1145/2398776.2398799

[11] Ajay Anil Mahimkar, Zihui Ge, Aman Shaikh, Jia Wang, Jennifer Yates, Yin Zhang, and Qi Zhao. 2009. Towards automated performance diagnosis in a large IPTV network. *ACM SIGCOMM Computer Communication Review* 39, 4 (2009), 231–242. Publisher: ACM New York, NY, USA.

[12] Hongzi Mao, Ravi Netravali, and Mohammad Alizadeh. 2017. Neural Adaptive Video Streaming with Pensieve. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication - SIGCOMM '17*. ACM Press, Los Angeles, CA, USA, 197–210. https://doi.org/10.1145/3098822.3098843

[13] Judea Pearl. 2009. *Causality*. Cambridge university press.

[14] Kenneth J. Rothman and Sander Greenland. 2005. Causation and causal inference in epidemiology. *American Journal of Public Health* 95 Suppl 1 (2005), S144–150. https://doi.org/10.2105/AJPH.2004.059204

[15] Rahul Singh, Prashant Shenoy, Maitreya Natu, Vaishali Sadaphal, and Harrick Vin. 2013. Analytical modeling for what-if analysis in complex cloud computing applications. *ACM SIGMETRICS Performance Evaluation Review* 40, 4 (April 2013), 53–62. https://doi.org/10.1145/2479942.2479949

[16] Kevin Spiteri, Rahul Urgaonkar, and Ramesh K. Sitaraman. 2016. BOLA: Near-optimal bitrate adaptation for online videos. In *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*. 1–9. https://doi.org/10.1109/INFOCOM.2016.7524428

[17] Mukarram Tariq, Amgad Zeitoun, Vytautas Valancius, Nick Feamster, and Mostafa Ammar. 2008. Answering what-if deployment and configuration questions with wise. In *Proceedings of the ACM SIGCOMM 2008 Conference on Data communication*. 99–110.

[18] Mukarram Bin Tariq, Murtaza Motiwala, Nick Feamster, and Mostafa Ammar. 2009. Detecting network neutrality violations with causal inference. In *Proceedings of the 5th International Conference on Emerging Networking Experiments and Technologies*. 289–300.

[19] Francis Y. Yan, Hudson Ayers, Chenzhi Zhu, Sadjad Fouladi, James Hong, Keyi Zhang, Philip Levis, and Keith Winstein. 2020. Learning in situ: a randomized experiment in video streaming. In *17th USENIX Symposium on Networked Systems Design and Implementation (NSDI 20)*. 495–511.

[20] Xiaoqi Yin, Abhishek Jindal, Vyas Sekar, and Bruno Sinopoli. 2015. A Control-Theoretic Approach for Dynamic Adaptive Video Streaming over HTTP. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication - SIGCOMM '15*. ACM Press, London, United Kingdom, 325–338. https://doi.org/10.1145/2785956.2787486