CIPAT: A Two-stage Configuration Impact Prediction Analysis Toolkit for Cellular Networks

KARTIK PATEL*, Chandra Department of Electrical and Computer Engineering, The University of Texas at Austin, Austin, United States
CHANGHAN GE†, Department of Computer Science, The University of Texas at Austin, Austin, United States
AJAY MAHIMKAR, AT&T Labs, Bedminster, United States
SANJAY SHAKKOTTAI, Chandra Department of Electrical and Computer Engineering, The University of Texas at Austin, Austin, United States
YUSEF SHAQALLE, AT&T Labs, Minneapolis, United States

Improving network performance and user experience by tuning network configurations is crucial to cellular service providers (CSPs). However, predicting the performance impacts of configuration changes is non-trivial. The large scale, diversity and complexity of configuration parameters and base station deployments, and more importantly, the uncontrollable external factors (*e.g.*, weather, called *latents*), lead to confounding effects between configurations and performance metrics. In this paper, we show that the effects of latents can be properly mitigated by considering intermediates, called *Mobility, Access, and Traffic (MAT) metrics*, which separate the configurations and latents from performance metrics. Then, we propose the **C**onfiguration **I**mpact **P**rediction **A**nalysis **T**oolkit (CIPAT), a novel two-stage toolkit, driven by a large real-world dataset from live LTE and 5G networks. Our extensive evaluation shows that CIPAT enables network operators to confidently predict the performance impact of candidate configuration settings with an accuracy of up to 86% and an efficacy of up to 85%.

CCS Concepts: • **Networks** → **Network performance modeling**; **Network management**; **Wireless access points, base stations and infrastructure**; • **Computing methodologies** → **Neural networks**.

Additional Key Words and Phrases: network performance modeling, network management, machine learning applications

1 Introduction

In cellular network management, tuning configuration parameters, such as signal power and handover thresholds, is one of the key tasks in improving the service performance. This task has been consistently challenging across different generations of cellular networks, from GSM (2G) to NR (5G), primarily due to the sheer number of parameters involved and their complex interactions. Cellular Service Providers (CSPs) have invested significant time and effort in searching for optimal configurations to ensure the best service. For example, in mid-size cities

*This work was conducted when Kartik Patel was a graduate student at the University of Texas at Austin. He is currently an assistant professor in the Department of Electrical Engineering at the Indian Institute of Technology Gandhinagar, Gujarat, India.
†This work was conducted when Changhan Ge was a graduate student at the University of Texas at Austin. He is currently a research engineer with AT&T Labs, Bedminster, NJ, United States of America.

Authors' Contact Information: Kartik Patel, Chandra Department of Electrical and Computer Engineering, The University of Texas at Austin, Austin, Texas, United States; e-mail: kartikpatel@utexas.edu; Changhan Ge, Department of Computer Science, The University of Texas at Austin, Austin, Texas, United States; e-mail: chge@utexas.edu; Ajay Mahimkar, AT&T Labs, Bedminster, New Jersey, United States; e-mail: mahimkar@att.com; Sanjay Shakkottai, Chandra Department of Electrical and Computer Engineering, The University of Texas at Austin, Austin, Texas, United States; e-mail: shakkott@austin.utexas.edu; Yusef Shaqalle, AT&T Labs, Minneapolis, Minnesota, United States; e-mail: yusef.shaqalle@att.com.

ACM 2376-3647/2026/4-ART
https://doi.org/10.1145/3811914

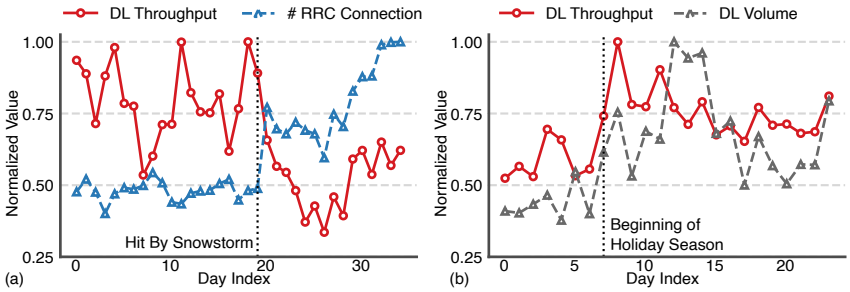


Fig. 1. Impact of latent variables on throughput (KPI). (a) snowstorm and (b) holiday

in North America, on an average, thousands of tuning attempts are made every week [31]. However, it remains nearly impossible to determine the performance impact of configuration changes a priori. In other words, the question faced by CSPs—"How will changes in configuration parameters impact Key Performance Indicators (KPIs)?"—remains unanswered without deploying changes on the network.

In principle, predicting the impact of configuration changes on KPI in such a complex inter-linked system requires a detailed *what-if* analysis, i.e., evaluating the potential effects of changes before they are applied. In addition to configurations, existing work [25] points out that the KPI is also related to a set of relatively static intrinsic attributes, such as morphology (urban/suburban/rural) and hardware. Hence, a naive approach could leverage large amounts of data to train a regression model mapping configurations and attributes to KPI. This is, however, significantly difficult in practice because other than configurations and attributes, the KPIs are also impacted by external factors such as weather conditions or events inducing large-scale changes in user dynamics. To illustrate, in Fig. 1, we show the impact of two different events, a snowstorm and a holiday break, on the throughput of an LTE cell in two different cities. In both cases, although the configuration parameters were not tuned when the event occurred, we still observe significant changes in the trends. Without prior knowledge of these events, i.e., purely based on the configurations and the attributes, it would appear that drastically different throughputs are observed for the same configuration. Therefore, these external factors, called *the latent variables*, bring challenges in accurately capturing the dependencies between configurations and KPI.

Even after decades of research, the problem of predicting the effect of a change in configuration parameter on the KPI in the presence of latent variables remains unsolved and continues to pose new challenges (see Section 2). In this paper, we show that the effects of the latent variables can be properly mitigated by using *observable* intermediate metrics, called *Mobility, Access, and Traffic (MAT) metrics*. These MAT metrics which include various network metrics such as average signal strength, traffic volume, and handover metrics, are routinely captured by CSPs as part of their daily operations. These MAT metrics have two important properties:

(a) The MAT metrics *isolate* both the configurations and the latent variables from the KPI variables (see Fig. 2). In Fig. 1, the number of Radio Resource Control (RRC) connections and Downlink (DL) volume are two examples of the MAT metrics. Irrespective of the event (snow, holiday), as long as such MAT metrics do not change, the KPI does not change. As an example, while the effect of transmit power (a configuration parameter) on throughput (a KPI) is influenced by snow, the relation between the Reference Signal Received Power (RSRP, a MAT metric) and throughput is not influenced by snow.

(b) The *direction of the change* (increase/decrease) in the MAT metrics can be predicted using the configurations even in the presence of the latent variables (see Section 7.2.1). For instance, increasing the transmission power (a configuration) will increase the average RSRP (a MAT metric), regardless of external events like a snowstorm. Similarly, reducing the handover threshold (a configuration) will increase the number of RRC connections (a MAT metric), even if there is weather anomaly.

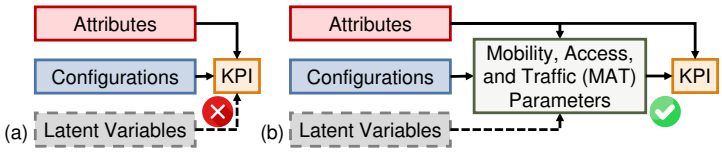


Fig. 2. Dependency graph. (a) One-stage. (b) Two-stage.

We emphasize that the MAT metrics are not isolated from latent variables. In fact, they are strongly dependent on them, but nevertheless, these metrics preserve the direction of change *w.r.t.* configurations in the presence of latent variables. This is a key observation that CSPs currently use while tuning the configurations. We exploit it for predicting the impact of configuration changes on MAT metrics, and subsequently on KPI, in the presence of latents.

Using the MAT metrics, we propose a two-stage toolkit called **C**onfiguration **I**mpact **P**rediction **A**nalysis **T**oolkit (CIPAT), which predicts the impact of configuration change on KPI in a latent-resilient[1] manner. The first stage of CIPAT predicts the direction (increase or decrease) of changes in the MAT metrics resulting from a proposed configuration change. Since the direction of change in MAT metrics is invariant to latent variables, this prediction can be made reliably without knowing the latent variables. The second stage of CIPAT uses a regression model to predict the change in KPI based on the potential values of MAT metrics. Since the impact of latent variables on KPIs is mediated by the MAT metrics, the regression model can be constructed using historical data without the knowledge of the latents and the coefficients of the model do not change with change in the underlying latents. Finally, CIPAT compares the likelihoods of KPI improvement and degradation and offers the operators a recommendation or a warning about the potential impact of the configuration change. Therefore, CIPAT, to the best of our knowledge, is the first latent-resilient toolkit to be able to reliably predict the impact of a configuration change on a live network, *before* deploying it on a live network.

Today, operators rely on their domain knowledge, experience, or engagement with equipment vendors to identify candidates for experimenting with new configurations on live networks, typically aiming to improve several KPIs. However, interactions with operators reveal that often the expectations are not met due to various external factors impacting the cellular network and only a small fraction of the changes are widely rolled out. Additionally, during the trials, any negative impact on performance requires timely identification and rollback. This is where CIPAT comes in: We envision CIPAT to be used as a filter by the network operators to proactively reject experiments of new configuration changes which could result in a KPI degradation.

CIPAT is driven by a large real-world dataset collected from a national CSP in the United States. It comprises daily cell-wise records of configurations, attributes, MAT metrics, and KPI measurements over the most recent 490 days from both LTE and 5G networks nationwide. Specifically, the LTE dataset is gathered from 1,525,404 cells, resulting in hundreds of millions of data points. The 5G dataset is gathered from 243,209 cells, resulting in over a million data points.

Our contributions: (1) We conduct the first-of-its-kind analysis of a large cellular network using a dataset from a major CSP in the United States. We show for the first time that the latent variables introduce challenges in a direct one-stage prediction of KPI changes from configuration changes. We then propose using intermediate metrics called *MAT metrics*. Due to the consistent direction of change in MATs *w.r.t.* configuration parameters in the presence of latents, and by creating isolation between KPI and the latents, these MAT metrics can serve as a bridge to measure the impact of configurations on KPI.

[1]A latent-resilient predictor is one which does not change with the state of the latent variables changes. Such predictors are strongly desired for various machine learning tasks, including causal inference and out-of-distribution generalization [2]. However, without strong assumptions on the underlying model, it is hard to assert true invariance. Since we are not proposing a formal model, our claim is more limited. Namely, our data analysis suggests that the two-stage pipeline mitigates the effects of latents, even if not truly invariant.

(2) We then propose CIPAT, a two-stage latent-invariant toolkit, to conduct what-if analysis for network operators. In the first stage of CIPAT, we use the state-of-the-art Feature Ordering by Conditional Independence (FOCI) algorithm [4] – a non-parametric feature selection algorithm – to identify the key configuration parameters. We then determine the direction of the impact that these parameters have on MAT metrics. In the second stage, we use regression models to predict the KPIs based on MAT metrics and attributes. Specifically, we design a novel Deep Neural Network (DNN) that can learn the attribute-dependent relationships between KPIs and MAT metrics.

(3) We present the benefits of CIPAT over the one-stage framework. We use both frameworks to predict the direction of the impact of throughput across multiple markets through a retrospective analysis. This involves studying past network configuration changes that lead to significant throughput variations, including instances that trigger rollbacks. Table 6 shows that CIPAT can detect configuration changes resulting in degradation with 70-85% accuracy and 78-86% efficacy, much higher than the accuracy (56-88%) and efficacy (52-78%) of one-stage models. Furthermore, CIPAT also shows a much higher efficacy (76-95%) in predicting the impact of untested configuration changes compared to the efficacy (51-75%) of the one-stage model.

2 Related work

2.1 Network planning and analysis

There is a rich literature focused on multidimensional aspects of network planning and analysis, such as detecting the performance change in the cellular network due to upgrades, configuration tuning, and minimizing the network outage [1, 5, 9, 11, 17, 22, 26, 27, 29, 36, 42–44, 46, 47]. Existing works in cellular configuration management focus on automating configuration tuning. Driven by a synthetic dataset, EXPLORA [12] enhances the explainability of Deep Reinforcement Learning (DRL)-driven control actions in Open RAN. Notably, CSPs are cautious about letting Artificial Intelligence (AI) directly take over the operation given the direct impact on customers and the practicality of decisions made by such AI agents [16]. Auric [28] recommends configurations based on the conformity to neighboring cell settings, while Aurora [25] is built on the top of Auric, enhancing recommendation quality by adding the conformity to "LMRD parameters" (a subset of the union of attributes and quantized MAT metrics) and performance-based filtering. Chroma [16] further correlates the impact of configuration change with "LMRD parameters" through rule learners, and hence recommends configuration changes that may potentially improve the performance. However, they regard some MAT metrics as relatively stable, which does not necessarily hold. Furthermore, none of these studies attempts to predict the performance impact. Only a few studies have attempted to predict the potential KPI impact [13] and the studies in the context of cellular networks typically use a limited set of configurations [40]. Moreover, none of these studies account for the impact of latent variables on KPIs or on the relationship between configurations and KPIs.

Our previous works [31, 32] are the first to analyze the impact on KPIs based on the observable parameters (configurations, attributes, and MAT) after accounting for external factors. In [32], we show that MAT metrics are superior to configuration parameters as KPI predictors amid latents. In [31], we introduce the initial concepts of the proposed toolkit CIPAT, detailing its construction and performance. In this paper, we expand them by offering data-driven case studies on CIPAT's foundational assumptions, highlighting its advantages, detailing construction stages with examples, and showcasing its operation in filtering the real-world configuration change on a live network.

2.2 Causal analysis in networks

To conduct a what-if analysis on a system requires conducting controlled experiments (aka interventions [33]) to avoid confounding where two variables appear to be correlated, but the correlation is actually due to the influence of a third variable. Without experiments, i.e., purely from the datasets, confounding effects can be avoided by

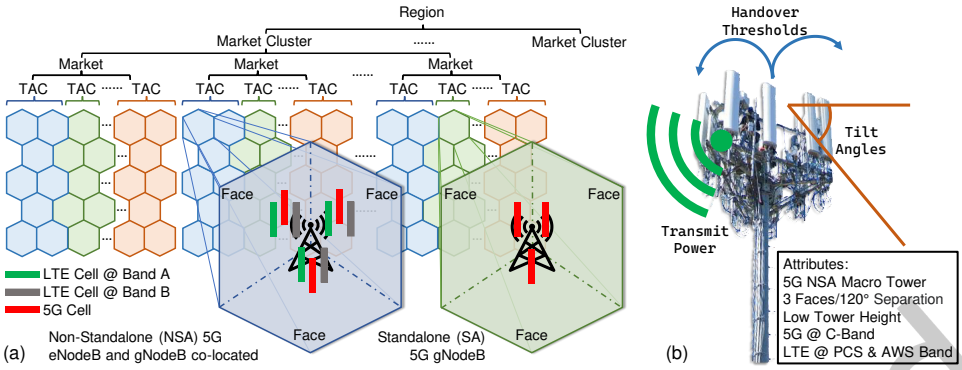


Fig. 3. Cellular network structure. (a) National topology. (b) An example 5G NSA base station.

finding and "conditioning" on a valid *adjustment set* in the dependency graph, effectively separating the influence of the rest of the variables from the target variable. This allows inferring the "true" effect of one variable on the other. If the dependency graph is fully known, we can find a valid adjustment and conduct a what-if analysis using only the dataset. These ideas lie at the core of causal inference [33, 35].

Many recent studies have focused on conducting a what-if analysis to determine the cause of failures or key parameters for optimizations in systems with unknown dependency graphs [15, 19, 23, 24, 37, 45]. They provide algorithms to efficiently search over possible causal graphical models. However, these studies are typically limited to settings where all relevant variables that could impact the performance of the network can be measured in the data.

The problem becomes more complex when the latent variables are present in the dependency graph [10, 38, 41]. Particularly, conducting a what-if analysis in a setting such as ours – where the underlying latent variables' characteristics and distributions can change over time – is still open after decades of research. If the latent variables had a fixed distribution, then alternate (but computationally intensive due to oracle access to Conditional Independence tests [6, 49]) approaches include [20, 34]. Furthermore, if bounds on the sensitivity of latent effects on treatment/outcomes were known, other approaches include [39, 48]. However, we are in a setting where neither of these hold – the latents are of vastly different types and not modeled by a single time-invariant distribution, and it is not clear how to pre-specify universal bounds on latent effects. Instead, our work leverages MAT metrics as mediators to mitigate the biases induced by latent variables. Different from the standard usage of mediators which involves estimating an end-to-end treatment effect (e.g., front-door adjustment [33]), we focus solely on estimating the direction of change in the first stage due to the specific properties of the MAT metrics as discussed above.

3 The dataset

3.1 Collection, organization, and notation

A national cellular network is partitioned into multiple regions, which are further partitioned into market clusters and markets. Each market, consisting of the base stations (BS; called *eNodeB* in LTE and *gNodeB* in 5G networks), is managed by a team of network engineers [28]. A BS has multiple *faces* that divide the space into different sectors. Each face operates multiple *cells*, potentially on different frequency bands, as well as over different technologies. A cell is the basic functional unit of a cellular network. As we mentioned in Section 1, we collect the daily snapshots of network data inventory from a national CSP in the US during the past 20 months. The data collected from each cell has 4 categories:

Attribute	Type	Example values	Attribute	Type	Example values
Region	Static	East, West	Morphology	Static	Rural, Suburban, Urban, Dense Urban
Markets	Static	***LTE***: 4 Market: NYC Manhattan, NYC East, AR, NM; ***5G***: Markets in NY, CA TX	Frequency Bands	Static	***LTE***: LTE-A < LTE-B < LTE-C < LTE-D < LTE-E; ***5G***: NR-A < NR-B < NR-C (mmWave)
Bandwidth	Static	5, 10, 15, 20 MHz	eNodeB Type	Static	Macro, Micro, Pico, iDAS, oDAS
Hardware	Static	***LTE***: HW_V1, 2, 3, 4, 5, 6(.1,2,3), 7(.1,2,3,4,5) ***5G***: HW_V1 (.1,2,$\cdots$,13), 2, 3(.1,2), 4(.1,2)	Software	Dynamic	***LTE***: SW_V1, 2(.1,2,3), 3, 4 ***5G***: SW_V1, 2(.1,2), 3(.1,2), 4, 5
DL MIMO Mode	Dynamic	Single Stream, Closed Loop MIMO	Tower Height	Static	Very Low, Low, Medium, High, Very High

Table 1. Attribute list of LTE and 5G datasets

(1) *Cell attributes*, such as BS type (*e.g.*, macro, pico), hardware, frequency, bandwidth, and morphology, define the operating characteristics of a cell. A list of attributes and their example values are listed in Table 1. The set of attributes is denoted by $\mathcal{A}$. The attribute values, indexed by date and cell, are denoted by matrix $\mathbf{A}$.

(2) *Configurations* of each cell such as transmitted power, antenna tilt angle, and handover thresholds are tuned by the network operators. The set of all configuration parameters (318 of them) is denoted by $\mathcal{C}$. The values of configuration parameters, indexed by date and cell, are denoted by the matrix $\mathbf{C}$.

(3) *MAT metrics* are monitored by CSPs for each cell every 15 minutes, then aggregated daily. They measure (i) User Equipments (UE) Mobility-related metrics such as UE distance and speed distributions, attempted number of Handover (HO) and success rates; (ii) Cell Access-related metrics such as average signal strength measurements; and (iii) Traffic metrics such as data volume, cell utilization and the number of UE connected to the cell. We consider a comprehensive set of 56 MAT metrics, denoted by $\mathcal{M}$. The values of MAT metrics, indexed by date and cell, is denoted by the matrix $\mathbf{M}$. A sample list of MAT metrics is given in Appendix A. We identify these MAT metrics by first listing commonly used metrics captured by CSPs in their day-to-day operations. We then filter the metrics using the Conditional Dependence Coefficient (CODEC) score [4] to remove the redundant metrics that have the least impact on KPI. Finally, through extensive discussions with CSPs, we select a subset of these metrics which indicates the latent-resilient direction of change *w.r.t.* configurations. While we believe the chosen MAT metrics cover a wide range of metrics, we recognize that these metrics may not be exhaustive and can be extended in future work.

(4) *Key Performance Indicators (KPI)* such as average per-user throughput, call retainability, and call accessibility[2] measure the quality of service experienced by the users. They are also monitored by the CSP for each cell every 15 minutes and then aggregated daily. The set of KPI is denoted by $\mathcal{Y}$. The values of KPI, indexed by date and cell, are denoted by the matrix $\mathbf{Y}$.

Notably, the dataset does not contain latent variables, such as weather history, network disruption, or large-scale external events (*e.g.*, sports) which can significantly affect KPI.

Notations: We denote a matrix of parameter set $\mathcal{A}$ by $\mathbf{A}$, its rows, indexed by i, by bold small letters $\mathbf{a}_i$ and columns, indexed by $i \in \mathcal{A}$, by $\mathbf{A}_i$. We denote the unique values of parameter set $\mathcal{A}$, by $\mathcal{U}(\mathcal{A})$. Finally, $sgn(\mathbf{A}) : \mathbb{R}^{m\times n} \rightarrow \{-1, 0, 1\}^{m\times n}$ denotes the element-wise sign function.

3.2 Derived features and latent variables

As discussed earlier, KPIs are not solely dependent on configuration parameters but are also influenced by various external factors such as weather conditions and large-scale events, and even the day of the week, which can significantly alter population mobility patterns. Among these external factors, it is easy to account for periodic variables such as the day of the week by explicitly adding them to the dataset. We refer to these as *derived features*.

However, exhaustively logging all events and quantifying their effects are difficult, if not impossible, tasks. We denote these unobserved external factors that can affect KPIs as *latent variables* in the dependency graph in Fig. 2. Without the data on latent variables, it is impossible to accurately model KPIs based on configuration

[2]In this context, accessibility denotes the network's ability to promptly provide resources for a UE, while retainability signifies the successful transition of a UE from one cell to another without dropping the connection.

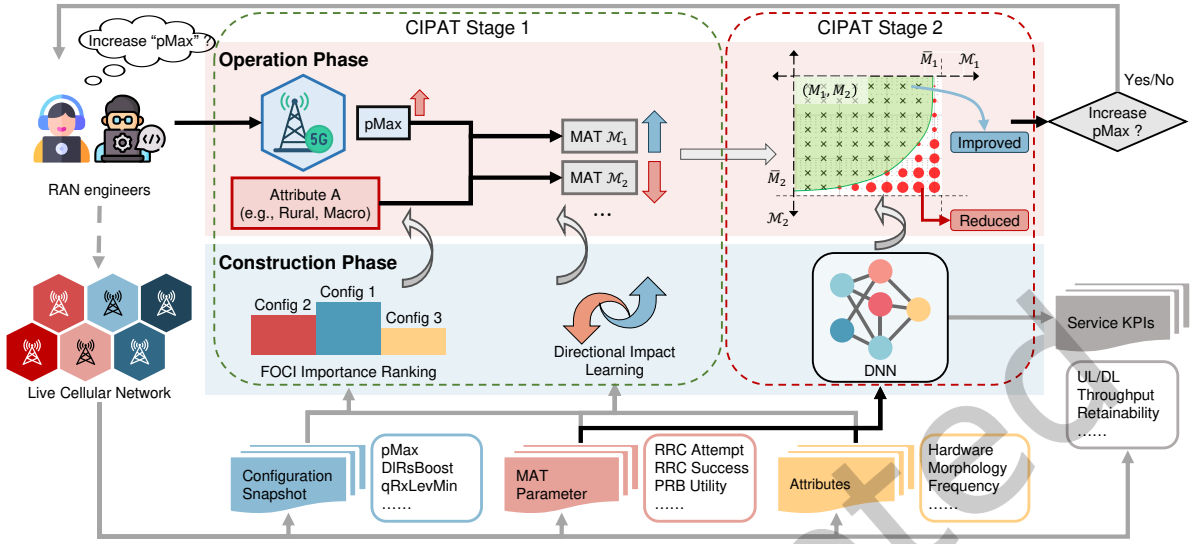


Fig. 4. The CIPAT workflow is divided into the construction phase and the operation phase. During the construction phase, CIPAT processes daily snapshots of the live cellular network to evaluate the significance of configuration parameters and understand their directional impact. Simultaneously, it trains a DNN to map MAT metrics to service KPIs. In the operation phase, CIPAT leverages the insights gained from the construction phase to assist RAN engineers in making informed configuration changes.

parameters, as any change in KPIs could be attributed to either changes in the configuration or the occurrence of such events. Additionally, conducting a controlled experiment on a national cellular network to assess the impact of configuration on KPIs is not practical due to the risk of affecting service quality for the users.

Therefore, we propose a two-stage framework CIPAT to predict the impact of configuration changes on KPIs without needing information on latents or conducting any controlled experiment.

4 CIPAT: design and operation

We present CIPAT, a novel two-stage toolkit designed for network operators to proactively determine the performance impact of configuration changes.

As shown in Fig. 4, When the operator wants to know the performance impact due to a change in a specific configuration parameter (*e.g.*, increasing the transmission power `pMax`) on a cell with attribute values **a**, CIPAT follows the following steps:

Stage 1: Configuration to MAT metrics. (1) First, CIPAT lists the MAT metrics known to be impacted by the chosen configuration parameter in cells with the attribute **a**. This list can be calculated in advance from the dataset using FOCI algorithm [4] (see Section 5.3). (2) Next, CIPAT identifies the direction of change in each of the MAT metrics due to the configuration change. This direction of change in MAT metrics can be estimated from the dataset without the knowledge of latent variables due to their resilience against underlying latents (see Section 5.4). The directions associated with each MAT metric present an orthant with the origin at current values of MAT metrics. Each point in the orthant denotes a possible value of the MAT metrics after the proposed configuration change.

Stage 2: MAT metrics to KPI. The second stage of CIPAT uses a pre-trained regression model (see Section 6) that maps (MAT, attributes) → (KPI). Then, the operators can use the model to predict KPI for different values of MAT metrics in the orthant and manually explore the trade-off between potential values of MATs (traffic,

coverage metrics) and KPI influenced by the proposed configuration change. The operators can also automatically estimate the potential value of the KPI at a large number of uniformly sampled points in the orthant. Depending on the volume of the orthant that predicts an increment/decrement in the KPI, the operators can decide whether to implement the proposed configuration change on the network.

The proposed two-stage process has additional benefits. First, instead of constructing the first stage solely from the dataset, we can also leverage domain-specific insights from the network operators. This modularity allows us to use both data-driven and domain-specific insights to predict the effect of a configuration change. This further enables the exploration of configurations that were under-tested but have the potential to have a significant boost.

Second, the two-stage process enhances the explainability of predictions. Even when a direct map of configurations to KPI was possible, it would have obscure important insights, while the two-stage method provides a mechanism to measure and predict the effect through the MAT metrics.

It is important to note that due to the presence of latents, we cannot predict the magnitude of the change in MAT metrics, which in turn prevents us from predicting the exact change in KPI. Instead, we are limited in estimating the direction of the change in MAT metrics and can only find the possible values of KPI for different values of MATs (can be explored by the operator or sampled from the joint empirical distribution of MAT metrics). This inability to map the configuration change to the KPI change can be considered the *cost of the latents*.

Note that CIPAT can be used to predict the impact of configurations on various cell-level KPIs, including but not limited to throughput, accessibility, and retainability. For the sake of brevity and focused analysis, our subsequent discussion predominantly centers on throughput as the selected KPI. In the subsequent sections 5 and 6, we describe the construction of each stage of CIPAT from the dataset. In Section 7, we validate three key properties of the MAT metrics that allows the construction of CIPAT. Finally, in Section 8, we provide the performance analysis of CIPAT.

One-stage alternatives for KPI prediction and justification for two-stage architecture. A one-stage model directly mapping (attributes, configurations, MAT metrics) $\rightarrow$ (the value of KPI) cannot be used to asses the impact of configuration changes on KPIs. This is because configuration changes affect MAT metrics, and the new values of MAT metrics after the configuration changes cannot be predicted due to unknown latent variables. Consequently, MAT metrics after the configuration changes are not available as input for such a one-stage KPI prediction model.

This fundamental limitation of the one-stage model motivates the two-stage architecture of CIPAT. The first stage of CIPAT models the impact of configuration changes on the *direction* of changes in the MAT metrics, a relationship known to be latent-resilient. The second stage then predicts KPIs based on the possible values of the MAT metrics after these configuration changes.

Alternatively, one could train a one-stage model mapping (attributes, configurations) $\rightarrow$ (KPI) using historical data. The hope for such model is that it implicitly generates rough estimates of MAT metrics by averaging over latents across many environments, which would then predict KPI. However, the predictions from such a single-stage model are only meaningful if the effects of latents are negligible. As we demonstrate in Section 7, this one-stage model performs poorly in our experiments, strongly indicating the non-negligible presence of latents in the dataset.

Finally, one could also envision a one-stage model mapping (attributes, current MAT metrics, configuration changes) $\rightarrow$ (direction of change in KPI). Training such a model, however, requires a dataset containing numerous configuration-change events for diverse values of configuration changes and MAT metrics, and their corresponding effects on KPI. In operational networks, configuration changes are infrequent compared to daily MAT/KPI logs, making such comprehensive data collection impossible without controlled experiments. Consequently, a unified model would suffer from severe data sparsity and poor generalization to unseen MAT metrics or novel configuration changes. CIPAT's two-stage design mitigates this issue by decoupling the learning tasks:

Stage 1 uses sparse configuration-change data to learn the direction of changes in MAT metrics, while Stage 2 leverages abundant MAT/KPI data to model KPI relationships. This decomposition makes the problem tractable and improves robustness to latent variables.

5 Constructing Stage 1: Configurations to MAT

In this section, we present the construction of the first stage of CIPAT: determining the relationship between configurations and MAT metrics from the dataset. In the following, we first outline the process of determining key configuration parameters for each MAT metric and describe the procedure for identifying the direction of change of MATs *w.r.t.* changes in configurations.

5.1 Mathematical setup.

We use CODEC [4] to statistically measure the conditional dependency between features (attribute or configuration) and the target variables (MAT metric or KPI). In the following, we first describe the CODEC statistic, followed by a greedy feature selection algorithm based on CODEC statistic.

A Measure of Conditional Dependence: Let the target variable and two mutually exclusive sets of features be denoted by x and P, Q. Then, the metric $T(x, P|Q)$, which measures the dependency between x and P conditioned Q, can be defined as follows [4]:

$$T(x, P|Q) = \frac{\int \mathbb{E}(\mathrm{Var}(\mathbb{P}(x \geq t|P, Q)|Q))d\mu(t)}{\int \mathbb{E}(\mathrm{Var}(\mathbf{1}\{x \geq t\}|Q))d\mu(t)}. \tag{1}$$

The quantity T can be interpreted as a generalization of the partial R^2 statistic that measures the proportion of variation in x that is explained by feature set (P, Q) but cannot be solely explained by the feature set Q [4]. Without loss of generality, we use the notation $T(x, P)$ to denote the measure of the "unconditional dependence" between the target x and the feature set P.

From a statistical perspective, $T(x, P|Q)$ represents a theoretical limit on how well a regression model can predict x based on P, for a given Q. If $T(x, P|Q)$ is *reasonably* high, then a good function approximator (such as a neural network) can find a function of P (for each value of Q) with a low prediction error. If $T(x, P|Q)$ is small, the variability of x cannot be captured adequately by a function of P alone (for any values of Q), thus, training a neural network is a futile exercise.

Estimating the quantity T with potentially continuous features in P, Q is computationally heavy. Thus, [4] proposes a consistent estimator of T, called **CODEC statistic** $\tilde{T}$, that can be calculated from the dataset in $O(n \log n)$ time-complexity where n denotes the number of samples in the dataset. For the complete procedure of calculating CODEC statistic, please refer to [4]. Note that there are a number of statistics that provide measures of conditional dependence between features and target variables [7, 21]. We use CODEC because it is a non-parametric, model-free statistic that does not require assumptions on the joint distribution of (x, P, Q). These benefits allow us to efficiently estimate conditional dependence in our large heterogeneous dataset containing discrete, continuous and categorical features. We present the difference between high and low-CODEC score between input features (i.e., configurations) and output (i.e., MAT metrics) in Section 5.2.

FOCI for Feature Selection: Feature Ordering by Conditional Independence (FOCI) [4] is a greedy approach for selection of the features based on CODEC statistic $\tilde{T}$. Specifically, it greedily collects features, one in each iteration, such that the CODEC score between collected features and the target is maximized. An outline of FOCI algorithm from [4] is reproduced in Algorithm 1.

5.2 Visualizing high and low impact configurations

We can visualize the CODEC score by contrasting the impact of two sets of configuration parameters on MAT metrics: those with high CODEC scores and those with low CODEC scores. Since the attribute values (e.g.,

Algorithm 1 FOCI algorithm (reproduced from [4])

Input: Dataset containing target x and feature set $\mathcal{F}$.
Let $K = |\mathcal{F}|$. Choose $G_1 = \arg\max_{u \in \mathcal{F}} \tilde{T}_n(x, u)$ and define $\mathcal{G} = \{G_1\}$
for $i = 2, ..., K$ **do**
  $G_i = \arg\max_{u \in \mathcal{F} \setminus \mathcal{G}} \tilde{T}_n(x, \{u\} \cup \mathcal{G})$
  if $\tilde{T}_n(x, \{G_i\} \cup \mathcal{G}) \leq \tilde{T}_n(x, \mathcal{G})$ **then**
    break
  $\mathcal{G} = \mathcal{G} \cup \{G_i\}$
Output: A set $\mathcal{G} = \{G_1, ..., G_{|\mathcal{G}|}\}$.

	Morphology	eNodeB Type	Hardware	Tower Height	Bandwidth	Frequency Band	Market	Software
Attribute Values a_1	Suburban	Macro	HW V1	Very High	10MHz	LTE-A	Arizona	SW V1
Attribute Values a_2	Urban			Very High		LTE-B	NYC East	
Attribute Values a_3	Suburban			Medium		LTE-A	NYC West	

Table 2. Attribute values a_1, a_2, a_3

markets or hardware) can affect the relation between the configurations and MAT metrics, we condition the dataset to select only those cells with specified attribute values given in Table 2[3].

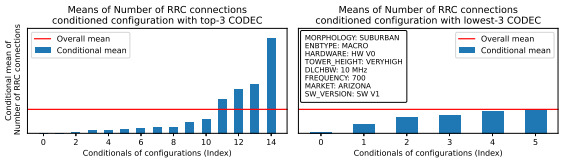
For the cells with each of the Attribute values: $\mathcal{A} = a_1$ (see Table 2), we calculate the corresponding CODEC scores $\tilde{T}_{(\mathcal{A}=a_1)}(M, c)$ between a MAT metric M and a configuration parameter $c, \forall c \in C$. We then sort to find the configuration parameters with the top-3 and bottom-3 CODEC scores, denoted by C_M^+, C_M^-, respectively. For instance, with the *number of RRC connections* as the MAT metric and $\mathcal{A} = a_1$, the top-3 and bottom-3 configuration parameters are C_M^+ = (`SnonIntrSearch`, `ThreshSrvLow`, `qRxLevMin`) and C_M^- = (`AdministrativeState`, `ActPDCCHLoadGen`,`DlInterferenceEnable`).

For a given MAT metric (and conditioned attribute values), a parameter that is among the 'top-3' will influence the MAT metric the strongest (and analogously the weakest for 'bottom-3'). We illustrate this in Fig. 5 generated by the following procedure: We consider top-3 configuration parameters for the number of RRC connections (a MAT metric) when conditioned on attribute values a_1 – (`SnonIntrSearch`, `ThreshSrvLow`, `qRxLevMin`). We then calculate the expected value of the number of RRC connections conditioned on `SnonIntrSearch` = c_1, `ThreshSrvLow` = c_2 and `qRxLevMin` = c_3, where $c_1 \in \{0, ..., 9\}, c_2 \in \{0, ..., 7\}, c_3 \in \{0, ..., 3\}$ are the possible values of the configuration parameters[4]. Then, in Fig. 5(a) (left), we plot the expected value of the number of RRC connections, conditioned on the configuration parameters having the values (c_1, c_2, c_3) indexed by $j \in \{0, ..., 14\}$[5]. A similar procedure is followed for bottom-3 configuration parameters, and the expected values are plotted in Fig. 5(a) (right). From Fig. 5, we can visualize the intuition behind the CODEC score: the MAT metrics are more sensitive to the values of the configuration parameters with higher CODEC scores than those with lower CODEC scores. Similar results can be observed in Fig. 5 when considering *average RSRP* as the MAT metric and considering the attribute values a_2 and a_3 defined in Table 2.

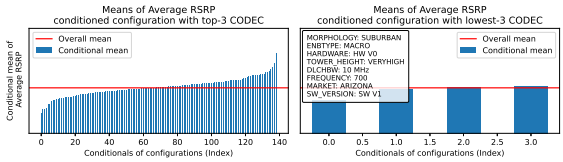
5.3 List of influential configurations.

Using the CODEC score, we now identify the most influential configuration parameters for a MAT metric on cells with attributes $\mathcal{A}$, by first selecting a conditional dataset given the attributes $\mathcal{A}$. Then we run FOCI algorithm [4]

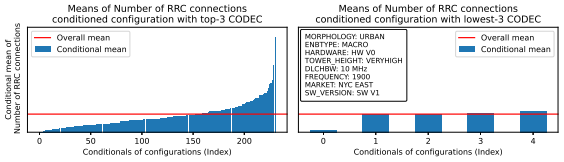
[3]Recall that the attributes do not have any parent nodes in the dependency graph, which allows conditioning the dataset on their attribute values without altering the graph dependencies (in effect, conditioning is equivalent to 'do-ing' an intervention in this case) [33].
[4]Ranges are normalized and discretized for the purpose of the illustration.
[5]Not all combinations of triplets exist in the dataset.



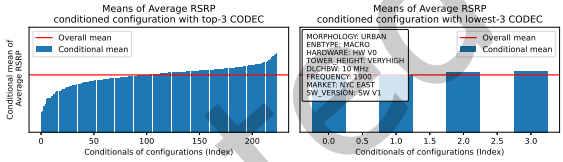
(a) Attribute a_1, Number of RRC sessions: the configuration parameters with top-3 and bottom-3 CODEC scores are (`SnonIntrSearch`,`ThreshSrvLow`,`qRxLevMin`), and (`AdministrativeState`,`ActPdcchLoadGen`,`DlInterferenceEnable`)



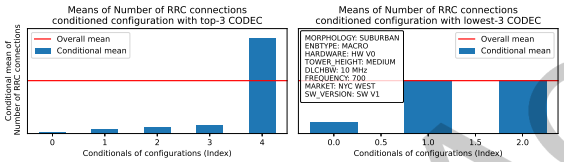
(b) Attribute a_1, Average RSRP: the configuration parameters with top-3 and bottom-3 CODEC scores are (`Angle`,`qRxLevMin`,`SnonIntrSearch`) and (`PrimPlmnCellRes`,`A5ReportInterval`,`ActPdcchLoadGen`)



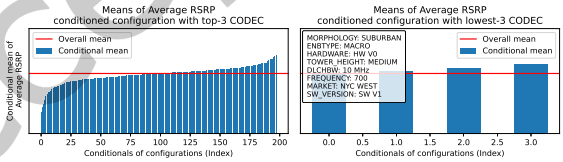
(c) Attribute a_2, Number of RRC sessions: the configuration parameters with top-3 and bottom-3 CODEC scores are (`Angle`,`pMax`,`ThreshSrvLow`), and (`MinDeltaRsrqIMLB`,`AdministrativeState`,`measBdw`)



(d) Attribute a_2, Average RSRP: the configuration parameters with top-3 and bottom-3 CODEC scores are (`Angle`,`ThreshSrvLow`,`pMax`) and (`EnableA3Event`,`iniPrbsUl`,`qOffFrq`)



(e) Attribute a_3, Number of RRC sessions: the configuration parameters with top-3 and bottom-3 CODEC scores are (`QRxLevMin`,`raNonDedPreamb`,`A3TriggerQuantityHO`), and (`AdministrativeState`,`PrimePlmnCellRes`,`ActTTI- Bundling`)



(f) Attribute a_3, Average RSRP: the configuration parameters with top-3 and bottom-3 CODEC scores are (`Angle`,`InterFrqThrH`,`InterFrqQThrHigh9`) and (`MaxNumUEDl`,`p0NomPucch`,`MechanicalAngle`)

Fig. 5. Conditional mean of MAT metrics against the values of the configuration parameters with top-3 and bottom-3 CODEC values for attribute value $\mathcal{A} = a_i, i = 1, 2, 3$

with a MAT metric M as a target and the configuration set C as the feature set. The FOCI algorithm iteratively collects the configurations in a way that maximizes CODEC score between the collected configurations and the MAT metric.

For example, Fig. 6 shows the important configurations identified by FOCI for the number of connections into a cell using the target variables: *(a)* number of RRC connections in LTE network and *(b)* number of UEs connections in 5G network. Specifically, we show collected configurations by FOCI at each iteration and associated cumulative CODEC score. The algorithm correctly identifies some of the most prominent configurations parameters. For instance, configurations related to cell coverage such as transmitted power (`pMax`) and antenna tilt (`Angle`), as well as handover-related configurations (`InterFreqThr`, `qRxLevMin-InterFreq` in LTE and `A1`, `A2`, `A3`, `A5` parameters in 5G) are among the most impactful configurations in networks. Furthermore, PHY layer parameters such as reference signal boost (`DlRsBoost`), and load balancing-related parameters (`IdleLBCapThresh`, `PDCCHLoadLevel` in LTE and `MaxNumOfUsersNRCell` in 5G) also impact the number of connections to a cell. Finally, derived periodic features such as day of the week also have significant impact on the number of connections.

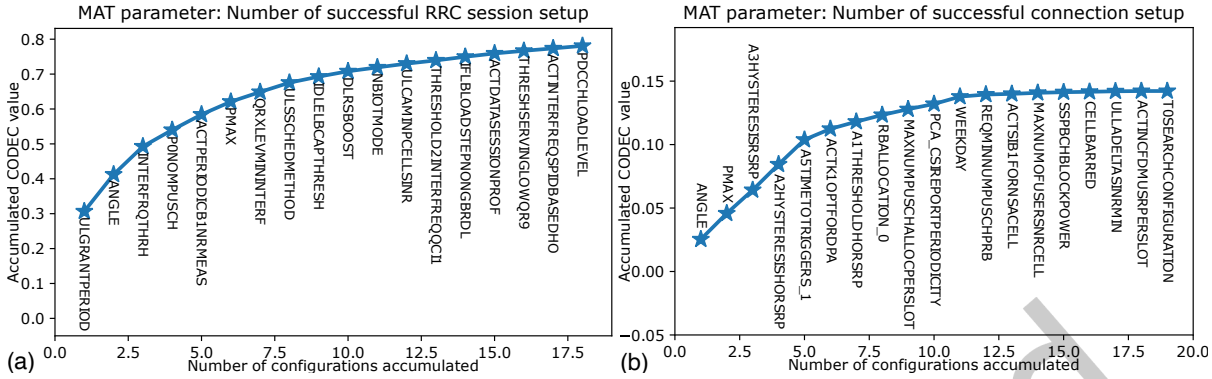


Fig. 6. Configurations identified by FOCI for # RRC connection setups in (a) LTE and (b) 5G network

We shared these results with the network operations teams, and confirmed that this aligns with their expectations and practices.

We note that the achieved CODEC scores in Fig. 6 are 0.15 for 5G and 0.75 for LTE, which falls short of the maximum possible value of 1, which means that there exists variability in MAT metric that is not explained by the configurations. This re-emphasizes that the latent variables have a significant impact on the MAT metrics. Thus, it may be hard to precisely predict the *value* of the MAT metrics, purely based on configuration parameters. However, in the next section, we show that the *direction of the change* in MAT metric due to the change in the configuration parameter can be predicted with sufficient accuracy.

5.4 Identifying the direction of change

Upon identifying the prominent influencers of all MAT metrics, we now determine if those configurations and MAT metrics change in the *same* direction (both increases or decreases together), or in *opposite* directions – increasing (or decreasing) configuration value decreases (or increases) the MAT metric. We first outline the dataset processing steps to identify the changes in configurations and associated changes in MAT metrics. Then, we describe the method for determining the direction of the change for each configuration-MAT metric pair with a fixed attribute value.

Data processing. We use daily snapshots of configurations, and computed the differences between the consecutive dates. We define a configuration change matrix $\Delta\mathbf{C}$, where $\Delta\mathbf{C}(i, c)$ represents the change in the configuration c in the i-th configuration change *event* (where i indexes the date and cell of this event). Given the configuration change matrix, the next step is to identify the direction of change (increase/decrease) in MAT metrics due for each configuration change. To do this, we compare the 7-day median value of MAT metrics before and after the change. Furthermore, we say that the MAT metric is unaffected by the change, if the relative change in the median before and after the configuration change is within a 3% margin. This allows us to avoid considering regular fluctuations in the MAT metric as a significant change. We denote the MAT change matrix by $\Delta\mathbf{M}$, where $\Delta\mathbf{M}(i, m) \in \{+1, -1, 0\}$ denotes the direction of the change in MAT metric m associated with i-th event.

Determining the attribute-wise direction of change. Finally, we estimate the direction of the change in MAT metric due to the change in the configuration parameter. We observe that the direction of change depends on the attribute of the cell. For instance, the tilt angle (a configuration) is a key influencer of the number of RRC connections (a MAT metric) in LTE network (as shown in Fig. 6). However, its impact on the RRC connection can vary depending on the morphology – rural sites might see increased connections with increasing tilt angle due to improved coverage, while urban sites might not have any effect of increasing tilt angle [16]. Hence, it

is essential to identify the direction of change as a function of attributes. In Section 8.4.2, we discuss the most relevant attributes that characterize the direction of change of different configuration-MAT pairs.

Following the above discussion, we measure the direction of change for each attribute value. We first partition the dataset by the attribute values of each cell associated with each configuration change event. Let $\mathbf{a} \in \mathcal{U}(\mathbf{A})$ denote an attribute value. Then for each data partition, we consider the associated subsets of $\Delta\mathbf{C}$ and $\Delta\mathbf{M}$, denoted by $\Delta\mathbf{C_a}$ and $\Delta\mathbf{M_a}$.

Moreover, several configurations may be adjusted simultaneously in practice [16]. To differentiate the effects of each individual configuration on MAT metrics, we use a linear model between $\Delta\mathbf{C_a} \rightarrow \Delta\mathbf{M_a}$. Although the relationship between them may not be linear and may not be latent-resilient, our goal is only to identify the direction of this relationship (*same* or *opposite*); hence, only the signs of the weights of the linear model suffice.

Formally, let the weight matrix $\mathbf{W_a}$ such that an element of the matrix, denoted as $\mathbf{W_a}(c, m)$, defines the impact of a configuration c on the MAT metric m for a given attribute value. Then,

$$\mathbf{W}^*_{\mathbf{a}} = sgn\left(\underset{W \in \mathbb{R}^{|C|\times|\mathcal{M}|}}{\arg\min} \|\Delta\mathbf{C_a}W - \Delta\mathbf{M_a}\|_2\right). \tag{2}$$

Here, $\mathbf{W}^*_{\mathbf{a}}(c, m)$ provides the relative direction of the change of the configuration-MAT metric pair for a given attribute values $\mathbf{a}$.

6 Constructing Stage 2: Predicting KPI from MAT

The Stage 2 of CIPAT maps the MAT metrics and attributes to KPI. Given that the impact of latents on KPI is significantly reduced due to the introduction of MAT metrics, we can train regression models to determine the relationship between attributes, MAT metrics and KPI. In this section, we discuss the design of the DNN model and training methodology used in CIPAT.

6.1 Design of DNN

Our DNN design is based on an observation that the relationship between MAT metrics and KPI varies across different network attributes. To achieve this, we use a shared representation for generalization and tunable heads tailored to specific attribute values. Specifically, the model takes MAT metrics $\mathcal{M}$ and attributes $\mathcal{A}$ as inputs and further categorizes attributes into two groups: *mask attributes* and *feature attributes*. Mask attributes define distinct relationships between KPI and MAT, necessitating separate model components. Feature attributes have a lesser impact and are treated as features. This classification is determined heuristically.

Mask attributes are one-hot encoded, and their combined representation is generated using the Kronecker product of their encodings, serving as a mask selector. The main branch processes MAT metrics and one-hot encoded feature attributes, consisting of 10 fully connected layers with leaky ReLU activations. The first 7 layers form the shared representation, which then branches into sub-networks for each mask attribute value. Each sub-network includes 2 hidden layers and an output layer, with the final output selected via the mask selector. Backpropagation collectively updates the shared representation weights and independently updates the corresponding sub-network weights. The DNN structure is illustrated in Fig. 7.

6.2 Training methodology.

We use two loss functions in training. Let $\mathbf{y}, \hat{\mathbf{y}}$ denote the true KPI and predicted KPI respectively. First, we use Mean Square Error (MSE) loss given by $\text{MSE}(\mathbf{y}, \hat{\mathbf{y}}) = ||\mathbf{y} - \hat{\mathbf{y}}||^2$.

Second, we use the Wasserstein distance as a loss to ensure that the predicted KPI conforms to the distribution of the ground truth [3, 8, 14, 30]. The Wasserstein loss is calculated as follows: Let $y_{(i)}$ denote i-th sample after sorting the vector $\mathbf{y}$. Then, $\mathcal{W}_1(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{n}\sum_{i=1}^{n} |y_{(i)} - \hat{y}_{(i)}|$. Jointly, the training loss function is given by

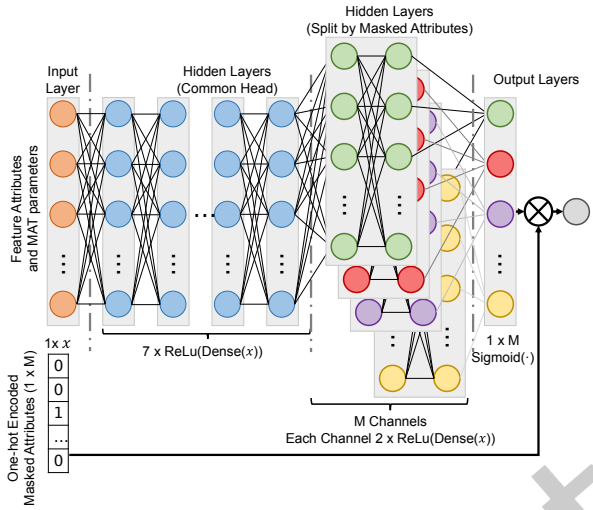


Fig. 7. A schematic of the proposed DNN

$\mathcal{L}(\mathbf{y}, \hat{\mathbf{y}}) = \text{MSE}(\mathbf{y}, \hat{\mathbf{y}}) + \beta \mathcal{W}_1(\mathbf{y}, \hat{\mathbf{y}})$, where we empirically choose $\beta = 0.5$. Finally, we use a standard MSE loss for evaluation of the regression model.

We choose 5×10^{-4} as the initial learning rate. To accelerate and smoothen the training process, we decrease the learning rate by a factor of 0.9 after the validation loss remains roughly unchanged for 5 consecutive epochs. We use the batch size of 256 data samples.

7 Validating key premises of CIPAT

Recall that there are three key pieces of information about MAT metrics used in CIPAT: *(a)* MAT metrics encode the latent variables, thus are better predictors of KPI. *(b)* The direction of change in MAT metrics *w.r.t.* the changes in configuration remains largely independent of the latents. *(c)* The relationship between MAT metrics and KPI also have minimal dependence on the latents. We validate each claim in the following sections.

7.1 Existence of latents and KPI prediction

One of the premises of our proposed two-stage framework is the fact that the one-stage map (configuration, attributes) $\mapsto$ (KPI) is not adequate at predicting KPI due to existence of latent variables. In this part, we validate this fact (1) statistically using CODEC score and (2) experimentally using DNN.

7.1.1 CODEC-based validation. In this case study, we statistically show that MAT metrics are better predictors of KPI compared to configurations. Specifically, for both LTE and 5G datasets, we consider each KPI as a target variable x, and measure the degree of dependence (conditioned on attributes) between KPI and:

(1) configurations, i.e., $P = \mathcal{C}$ and $Q = \mathcal{A}$.
(2) MAT metrics, i.e., $P = \mathcal{M}$ and $Q = \mathcal{A}$.

Table 3 shows the estimated CODEC values. We observe that when the set P includes MAT metrics, the estimated CODEC values are higher compared to when set P only includes configurations. This means that MAT metrics encode more information about KPI by encoding the latent variables which configuration parameters cannot (see Fig. 2). Therefore, we can conclude that MAT metrics can better predict KPIs compared to configurations.

LTE KPI x	Configuration $P = C$	MAT $P = \mathcal{M}$	NR KPI x	Configuration $P = C$	MAT $P = \mathcal{M}$
DL Throughput	0.2323	**0.6176**	DL Throughput	0.0959	**0.6641**
Retainability	0.2999	**0.4093**	Retainability	0.0512	**0.4613**

Table 3. CODEC score $\tilde{T}(x, P|\mathcal{A})$.

7.1.2 DNN-based validation. In this case study, we empirically validate that MAT metrics are better predictors of KPI compared to configurations. We consider the LTE dataset, and use a set of all configuration and DL throughput (as a KPI) for all date and cell combinations, excluding any configurations that remain constant throughout the dataset. For the one-stage prediction, we employ a neural network similar to the one described in Section 6.1, but with 152 neurons per layer to accommodate increased number of inputs (number of configurations > number of MAT metrics). The market, morphology and tower height are designated as *mask attributes*, while hardware, software, enodeB type, frequency, MIMO mode and weekdays are considered as *feature attributes*. The model is trained using the same methodology as outlined in Section 6.2. The achieved MSE of **One-stage** (Configurations to KPI) model is 0.00638, while the MSE of **Two-stage** (MAT metrics to KPI) model is 0.00197 (see Table 4). Evidently using MAT metrics and attributes yields improved DL throughput.

Takeaway. *Higher CODEC score and lower prediction error using DNN model validate the fact that MAT metrics are better predictors of KPI than configurations.*

7.2 Impact of latents

7.2.1 On the directions of changes in MAT metrics. Operators use prior knowledge about the directions of change (increase/decrease) in various metrics when making changes to a well-explored configuration, which subsequently provides the basis for the prediction in the direction of KPI change. The direction of change in many of these metrics generally does not depend on the underlying latents. We designate such metrics as MAT metrics. In this part, we give data-driven case studies of a few MAT metrics and show that the change in MAT metrics *w.r.t.* configuration changes is consistent considering tourism and foliage as latent variables.

Case Study 1.1: Tourism: We study a market of a tourism destination that has high traffic volume and user activity during the peak tourism season, but these activities significantly drop in the off season. We use the LTE dataset of this market, with attributes, configurations and two MAT metrics – the number of RRC connections and the average RSRP. We split the dataset into two partitions, one containing data from the peak tourism season and the other from the low tourism season. For each partition, we follow the process described in Section 5 for both MAT metrics: Determine the direction of change in the MAT due to changes in the key configurations impacting the MAT metrics. To measure 'similarity' between the seasons, we calculate the fraction of attribute values for which the estimated directions of change associated with a configuration were similar.

We find that for the number of RRC connections, the direction of change due to the change in the transmission power remains consistent across both seasons for 80% of attribute values. Similarly, in the case of average RSRP as the MAT metric, this metric reaches 90%. This validates the idea that the direction of change in the MAT metric due to a configuration change is consistent regardless of the change in the underlying latent variable.

Case Study 1.2: Foliage: We perform a similar study using the dataset from a market where there is a noticeable difference in foliage between summer and winter seasons. The presence/absence of leaves on trees significantly alters the propagation environment; furthermore, the exact date of foliage loss is highly location specific, and thus is hard to explicitly account for without considerable effort. Thus, considering 'foliage' as a latent variable, we use two subsets of LTE dataset of the market: One from summer during the peak vegetation, and second from winter after the fall season.

Similar to the previous study, we find that for both the number of RRC sessions and the average RSRP, the direction of change associated with the change in the transmission power remains the same across both seasons for 90% of all attribute values.

Takeaway. *Existing heuristics used by the operators for configuration tuning assume that the directions of change in MAT metrics* w.r.t. *configuration changes do not depend on the underlying latents. We validate this by taking examples of two known latents and their changes and show that the direction of change in the MAT metric due to a configuration change remains consistent even with the change in the underlying latent variables.*

7.2.2 On MAT - KPI relationship. We now do similar case studies to illustrate that the relationship between MAT and KPI is robust to changes in the latents.

Case Study 2.1: Tourism: Similar to the case study 1.1, we study the tourism-dependent market and consider the 'tourism activity' as the latent. Our observation is that a predictor of KPI trained on attributes and MAT metrics from the peak tourism season data can accurately predict KPI for the other season and vice versa, indicating the relationship between MAT and KPI is approximately invariant to this latent.

We use the LTE dataset of attributes, MAT metrics and DL throughput (a KPI) of all cells in the market and split it into two partitions: the dataset $\mathcal{D}_1$ corresponding to the peak tourism season, and the dataset $\mathcal{D}_2$ associated with the rest of the period. We then train two neural networks $\mathcal{N}_1, \mathcal{N}_2$ using the training partitions of $\mathcal{D}_1, \mathcal{D}_2$, respectively. This results in each network being accurate predictors of KPI (achieved MSE $\sim$ 0.0015) during their respective two seasons (the network is similar to the DNN in Section 6).

Now, to verify the "similarity" in the regression models, we calculate the sample-wise Normalized MSE (MSE) metric as follows: Let $\mathbf{x} \in \mathcal{D}$ denote one sample of attributes and MAT metrics from a dataset $\mathcal{D}$. Then,

$$\text{NMSE} = \mathbb{E}_{\mathbf{x} \in \mathcal{D}_1 \cup \mathcal{D}_2} \left[\frac{(\mathcal{N}_1(\mathbf{x}) - \mathcal{N}_2(\mathbf{x}))^2}{(\mathcal{N}_1(\mathbf{x}) + \mathcal{N}_2(\mathbf{x}))/2} \right]. \tag{3}$$

This metric measures the *sample-wise* similarity in the predictions made by two separately trained neural networks, and is empirically computed using holdout data (meaning data not used for training). For this case study, the measured *NMSE is 0.0001* (0.01% prediction difference), thus showing that both neural networks can accurately predict the samples from the alternate dataset.

Case Study 2.2: Foliage: We consider the same market used in the study 1.2, which observes significant foliage. We consider the 'foliage' as a latent variable and use two subsets of the dataset: first from summer during the peak vegetation, and second from winter in the considered markets. Similar to the study 2.1, we train two predictors based on attributes and MAT metrics using these two different datasets, achieving MSE $\sim$0.002 and resulting in accurate predictors of KPI. We then compute NMSE (3) and found it to be 0.0002 (or 0.02% prediction difference).

Takeaway. *Using two known latents in two markets, we show that the DNNs trained with different underlying latent states yield the same predictions when testing them under another latent state. This anecdotally verifies that the relationship between the KPI and MAT metrics seems roughly invariant to latent effects.*

We recognize that our results indicating minimal effects of latents on the relationship between MAT and KPI (and the directions of change in the previous section) are anecdotal and dependent on the chosen latent variable and datasets. Indeed there are 'known latents' (which we have used in these studies), but there are a lot more 'unknown latents'; thus it is impossible to definitively prove isolation of KPI from the latents. Nevertheless, based on field knowledge from operators combined with these case studies, we move forward with the results.

8 Evaluation

In the following section, we provide the performance results related to CIPAT. We first optimize the performance of the MAT to KPI model by tuning the mask attributes needed to accurately characterize MAT-KPI relationship. We then show the detailed operation of CIPAT using an example of a configuration change on a live network. We

Model	Mask/Feature Attributes (M/F)					MSE	wrt Variance = 0.015	
	DL MIMO Mode	Frequency	eNodeB Type	Hardware	Software		Residual Error	Improvement
One-stage	M	F	F	F	F	0.00638	42.54%	N/A
CIPAT Two-stage (DNN)	F	F	F	F	F	0.00185	**12.34%**	**+30.20%**
	M	F	F	F	F	0.00197	13.10%	+29.44%
	M	M	F	F	F	0.00218	14.53%	+28.01%
	M	M	M	F	F	0.00204	13.62%	+28.92%
	M	M	M	M	M	0.00197	13.16%	+29.38%
Common Mask Attributes	Market, Morphology, Tower Height							
Common Feature Attributes	Weekday, Bandwidth							

Table 4. Achieved MSE in predicting DL throughput by models jointly trained across 4 markets of LTE network with varied mask attributes.

then evaluate the performance of CIPAT showing its effectiveness as a pre-filter for the operators. Finally, we provide additional results and insights into CIPAT.

For the evaluation of CIPAT, we primarily use LTE dataset due to a large number of configuration changes in the dataset. We consider the attributes listed in Table 1, 189 out of 318 configurations which were changed at least once over 490 days and 56 MAT metrics (as listed in Appendix A). We set DL throughput as the target KPI.

8.1 Tuning the mask attributes

To achieve optimal performance within the proposed DNN model, we explore the precise tuning of mask attributes. Mask attributes offer a mechanism for fine-tuning the model, enabling model optimization tailored to specific operational scenarios. To this end, we experiment with the combinations of mask and feature attributes and summarize the resultant MSE errors in Table 4.

While in theory, an increase in mask attributes could lead to highly refined models, our empirical findings suggest a more nuanced relationship between mask attributes and performance. We observe a consistent increase in MSE error with each additional mask attribute. This can be attributed to a reduction in the data samples available for fine-tuning the head layers, as the number of mask attributes increases. Therefore, the indiscriminate addition of mask attributes, as demonstrated in our experiments, may not yield the desired performance gains. Instead, a judicious mask attribute selection that prioritizes attributes with the most significant influence on the relationship between MAT metrics and KPI is essential for achieving optimal results.

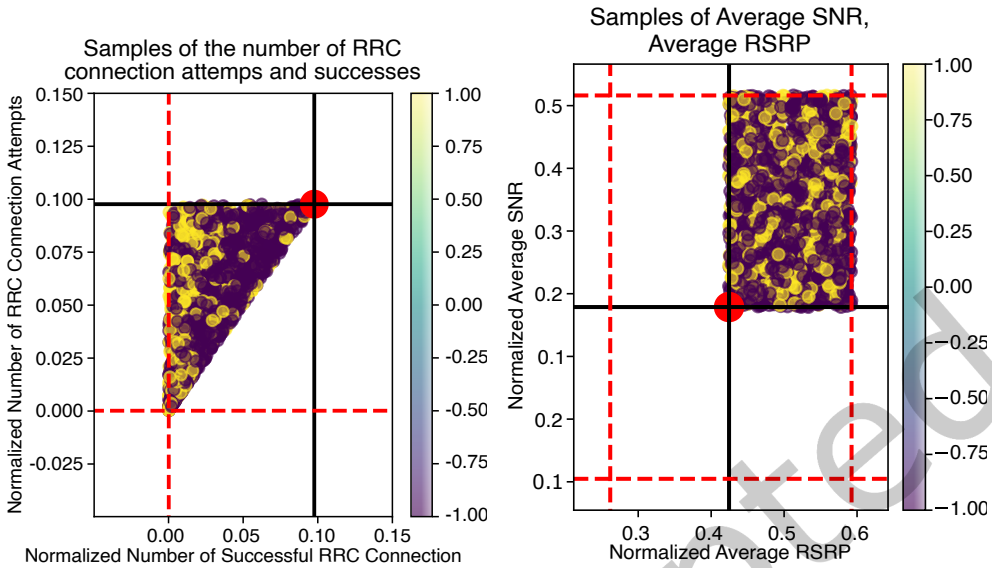
8.2 Operating CIPAT: An Example

We further give a working example of CIPAT from the historical data. Consider the cell X from Market A. From the dataset, we observe that the operator attempted to increase the tilt angle from 20° to 40°. If CIPAT had been used as a filter, the operator would have provided it with three sets of values: (1) The cell name X, (2) the configuration and new value, i.e., the angle from 20° to 40°, and (3) the median of MAT metrics over last 7-days.

In the first step, CIPAT finds the attributes of the cell from a lookup table. The identified attributes of cell X is given in Table 5. The toolkit then uses the given cell attributes and finds the directions of change in all MAT metrics for the change in the tilt angle. These directions can be obtained from the pre-calculated weight matrix $\mathbf{W}^*_{\mathbf{a}}(\texttt{ANGLE}, \cdot)$ given in (2).

These directions associated with all 56 MAT metrics present a 56-dimensional orthant with the origin at current values of MAT metrics. This orthant is the space of all combinations of the possible MAT metric values after the proposed configuration change. We sample 2000 points from this orthant and plot their 2D projections in Fig. 8.

Each sampled point is then passed through the trained regression model to get an estimated direction of change in cell throughput. In Fig. 8, we denote the positive and negative directions of change in cell throughput



(a) (b)

Fig. 8. A 2D projection of 56-dimensional orthant on two-pairs of MAT metrics: The dotted red lines shows the historical limits of the MAT metrics associated to the identified attributes. The intersection of black lines show the current 7-day median of the MAT metrics. Each point (colored dot) represents a sampled combination of possible MAT metric values resulting from the configuration change. The color of the dots show the predicted impact on KPI. There are more points with negative impact on KPI (~72%) than with positive impact on KPI.

Morphology	**eNodeB Type**	**Hardware**	**Tower Height**
Suburban	Macro	HW V1	Medium
Bandwidth	**Frequency Band**	**DLMIMO Mode**	**Software**
10MHz	LTE-D	Closed Loop MIMO (4x4)	SW V1

Table 5. Attribute values of cell X from Market A

with blue and yellow colors. Finally, given the higher number of points with a negative expected change in cell throughput, CIPAT concludes that increasing the angle on the cell X can negatively impact cell throughput. With the suggestion by CIPAT, the operator can put the configuration change on hold. The operator can also further investigate the result from CIPAT by observing various projections of the orthant (similar to Fig. 8), and use their knowledge to overrule the decision.

8.3 Performance of CIPAT

In this section, we present a detailed evaluation of CIPAT through post-facto analysis. CIPAT is designed to act as a safeguard for operators, ensuring configuration changes do not degrade cell-level KPIs before being applied on live networks. Accordingly, its effectiveness is measured using two key criteria:

(1) *Accuracy:* CIPAT should identify all configuration changes that degrade throughput. High accuracy indicates the ability to filter most throughput-degrading changes.
(2) *Efficacy:* CIPAT should only suspect those changes that result in degradation. High efficacy ensures minimal false positives, filtering only genuinely degrading changes.

Accuracy and efficacy correspond to recall and precision, respectively, if we treat CIPAT as a binary classifier. Let $\mathcal{D}$ denote observed degradation and $\hat{\mathcal{D}}$ denote predicted degradation. Then, the accuracy can be defined as $\mathbb{P}(\hat{\mathcal{D}}|\mathcal{D})$, while the efficacy can be defined $\mathbb{P}(\mathcal{D}|\hat{\mathcal{D}})$.

8.3.1 Methodology: We evaluate CIPAT on four U.S. markets, including urban, suburban, and rural regions with varied geography. We construct CIPAT using 260 consecutive days of cell-level metrics per market and follow procedures from Sections 5 and 6 for each market. Then, we use the constructed toolkit to evaluate the impact of configuration changes on cell-level throughput.

We consider the data from the subsequent 230 consecutive days as the test dataset. In practice, the operators are primarily interested in filtering the configuration changes that can significantly degrade a KPI – minor degradation in one KPI is tolerable to explore improvement in other KPIs. Therefore, we consider evaluation of CIPAT for the configuration changes in the test dataset that meet two criteria: (1) there was at least a 10% change (improvement or degradation) in cell-level throughput due to the configuration change, and (2) there was at least a 10% change (improvement or degradation) in eNodeB-level throughput.

We pass each cell-date pair in the test set with its configuration change and the cell attribute through CIPAT according to the process described in Section 8.2. The first stage predicts the direction of the change in MAT metrics based on each configuration change and the cell attributes, generating an orthant constrained by two conditions: *(a)* the number of attempts of the mobility management related procedures (*e.g.*, handovers, RRC session setup) must be higher than the number of successes, and (2) the range of each MAT metric does not exceed the historically observed ranges for the given attributes. CIPAT then samples 20,000 potential MAT metric values from the orthant. The second-stage DNN predicts throughput for each sample, determining the directional change via majority voting. Accuracy and efficacy are calculated by comparing predictions against observed outcomes from the test dataset.

8.3.2 Accuracy of CIPAT ($\mathbb{P}(\hat{\mathcal{D}}|\mathcal{D})$). We measure the accuracy of CIPAT by counting all configuration changes from the test data that are correctly identified as degrading throughput and dividing it by the total number of degradations observed in the test data. Results (see Table 6) show accuracy reaching up to 85%, though performance varies across markets. For instance, Market D exhibits lower accuracy (70%) due to its diverse landscapes, including dense urban areas and sparsely populated regions. Potential improvements include using hourly MAT metrics and incorporating neighboring-cell impacts, as discussed in Section 9.

8.3.3 Efficacy of CIPAT ($\mathbb{P}(\mathcal{D}|\hat{\mathcal{D}})$). Efficacy assesses CIPAT's reliability in predicting severe degradations, particularly those exceeding 10% eNodeB-level throughput decline. We measure the efficacy by calculating the fraction of actual degradations out of all predicted degradations. We present the efficacy results in Table 6.

8.3.4 Efficacy of CIPAT in predicting the impact of untested configuration changes. A key challenge for the operators is predicting the impact of untested configuration changes. A configuration is considered "novel" if it has not been applied to cells with identical attributes. Table 6 highlights CIPAT's performance in tackling this vital challenge, demonstrating its ability to generalize predictions to untested scenarios.

8.3.5 Comparison with one-stage model. We compare CIPAT against a one-stage model that directly predicts KPI degradation based on configuration changes and attributes. As presented in Table 6, CIPAT outperformed the one-stage model by 10-40%. Moreover, CIPAT excelled in handling novel configuration changes, which can be viewed as "out-of-distribution" samples, meaning that the test environment requires the model to predict about scenarios not present in the training dataset. This phenomenon aligns with a well-established principle in causal inference: understanding relationships prevents the learning algorithm from fitting models to spurious correlations, thereby enhancing model performance on out-of-distribution samples. By explicitly modeling

Market & Market Attributes	Accuracy: $\mathbb{P}(\hat{\mathcal{D}}\|\mathcal{D})$		Efficacy: $\mathbb{P}(\mathcal{D}\|\hat{\mathcal{D}})$			
	CIPAT	One stage	All changes (CIPAT)	All changes (One-stage)	Novel changes (CIPAT)	Novel changes (One-stage)
(A) Densely populated metropolitan downtown, plains	**0.85**	0.62	**0.82**	0.71	**0.95**	0.67
(B) Densely populated suburban, plains	0.74	**0.88**	**0.86**	0.78	**0.93**	0.75
(C) Densely populated urban and sparsely populated mountains and deserts	**0.70**	0.67	**0.78**	0.53	**0.76**	0.54
(D) Sparsely populated regions with mountains, plateaus, and deserts	**0.78**	0.56	**0.78**	0.52	**0.81**	0.51

Table 6. Accuracy and efficacy of CIPAT and one-stage model across markets: CIPAT performs better than the one-stage model, particularly in markets with significant seasonal user dynamics. This highlights the benefit of the two-stage approach using MAT metrics in addressing the impact of underlying latent variables.

relationships between configurations and KPIs via MAT metrics, CIPAT mitigates spurious correlations that often impair one-stage models. This advantage is empirically evident from the results in Table 6.

We also note that, in Market B, the one-stage model attains comparatively higher accuracy. While our current dataset does not allow us to isolate the cause, we find two plausible explanations consistent with our overall analysis. First, Market B contains the largest number of observed samples, which may allow the one-stage model to fit market-specific patterns more easily. Second, if latent variability in Market B is lower or more homogeneous than in other markets, the direct configuration-to-KPI mapping learned by the one-stage model, especially with a large number of samples, may experience fewer violations due to unobserved external factors. However, such effects also increase the risk that the one-stage model overfits correlations as implied by its worse performance on *novel* configuration changes. In contrast, CIPAT maintains consistently high efficacy across markets, particularly for *novel* configuration changes, indicating that the two-stage design captures underlying directional relationships rather than coincidental market-specific correlations. A deeper investigation of these market-level differences would require more granular latent information or controlled configuration tests, and we leave this as an important direction for future work.

8.3.6 Runtime of CIPAT.

Training time of CIPAT. Constructing CIPAT involves constructing the two stages of CIPAT. The construction of Stage 1 is dominated by the FOCI algorithm, whose per-iteration cost is $O(nC \log n)$, where n denotes the number of samples in the dataset, and C denotes the number of configuration parameters. As an example, for Market D (900,000 samples; 114 configurations with non-unique values), identifying the top-20 influential configurations for a MAT metric required around 91.8 CPU-hours (1.53 hours using 60 parallel threads). The construction of Stage 2 (DNN training) grows with $O(n)$, which translates to 4 hours of training for Market D on a single core of Intel Xeon Gold 6152 cluster.

Inference time of CIPAT. Once the CIPAT models are trained, the inference step of the first stage (i.e., predicting the direction of MAT metrics for a queried configuration change) only involves a search in a lookup table, which has a constant computational complexity. The inference step of the second stage, however, requires evaluating potential KPI outcomes from a large number of values of the MAT metrics sampled from the orthant chosen by the first stage. This DNN-based prediction task has linear computational complexity with respect to the number of sampled values of MAT metrics. Therefore, the overall runtime of CIPAT is primarily determined by the number of points sampled from the chosen orthant. In our analysis, the average end-to-end inference time for a trained CIPAT was 1.19 seconds for 20,000 sampled MAT metrics on a single core performance on an Intel Xeon Gold 6152 cluster.

Takeaway. *Post-facto analysis on the real-world configuration tuning dataset demonstrates that CIPAT can identify up to 85% of configuration changes that cause performance degradation. Furthermore, with 78-86% confidence, CIPAT's prediction of throughput degradation correlates with an actual decline. Importantly, CIPAT can identify the impact of untested configuration changes with up to 95% efficacy. Compared to a one-stage tool that only uses configurations and attributes for KPI prediction, CIPAT is more accurate and effective toolkit to filter the potentially damaging configuration changes before implementing on a live network.*

8.4 Additional Results and Insights

In this section, we provide a few additional insights derived from our analysis. Specifically, we provide a comparison between the key configuration parameters in LTE and 5G that impact the number of connections to a cell, and then, discuss the attributes that have the most impact on the direction of the change of MAT metrics.

8.4.1 Impact of configurations on MAT: Comparing LTE and 5G. To analyze critical differences across the LTE and 5G networks, we categorize the configurations selected by FOCI into the following 7 categories: (1) *Operational mode* - such as the support for NB-IoT or Category-M devices; (2) *Coverage related configurations*, *e.g.*, tilt angle and power; (3) *Handover related configurations* used for transferring active sessions across cells; (4) *MAC and network layer configurations*, *e.g.*, scheduler parameters, and limits on the number of sessions and resource block allocations; (5) *PHY layer configurations* related to MIMO configuration, carrier aggregation and channel state information feedback; (6) *Idle mode configurations* that define the transition from active session to idle mode and cell search and inter-cell transitions during the idle mode; (7) *Periodic features* (derived features such as day of the week).

In Fig. 9, we plot the number of occurrence (normalized) of these categories in top configurations for number of connections to a cell as identified by FOCI. We observe that the number of connections to an LTE cell is significantly dependent on idle mode parameters as compared to other categories. However, in the 5G network, the MAC and network layer parameters along with handover parameters have a significant impact on number of connections. This observation can be explained by the fact that the current deployments prioritize the active UEs to connect to the 5G network. Therefore, the network and MAC parameters which defines the limits on the number of sessions and resource block allocations, and handover parameters which controls the transfers of active sessions, have more impact on the 5G network.

8.4.2 Key attributes affecting the direction of change in MAT metrics. In Section 5.4, we defined the matrix $\mathbf{W}^*_{\mathbf{a}}(c, m)$, $\forall a \in \mathcal{U}(\mathbf{A})$, denoting the direction of change for a given configuration $c \in C$ (e.g. tilt angle) and MAT metric $m \in \mathcal{M}$ (e.g. number of connections) pair. In this section, we find the attributes that characterize the direction of the change of each configuration-MAT metric pair.

Let $\mathbf{a}$ be a row-vector denoting a value of the attributes in set $\mathcal{A}$. By concatenating all attribute values, a matrix $\mathcal{U}(\mathbf{A})$ with features $\mathcal{A}$ can be generated. Similarly, we define a direction-vector $\mathbf{W}^*_+(c, m)$ containing the direction of the change of all attributes. We then run the FOCI algorithm (Algorithm 1) with $\mathbf{W}^*_+(c, m), \forall \mathbf{a}$ as a target and $\mathcal{A}$ as a feature set. The results of FOCI indicate the most prominent attributes that characterize the direction of change in MAT metrics due to the change in configuration c.

We then show a comparison between these key attributes of two different MAT metrics. Specifically, for each MAT metric, we count the number of times each attribute is listed as the top-3 contributor (based on CODEC score) across configuration parameters. We then plot in Fig. 10, a normalized count of each attribute for two MAT metrics: (1) Number of RRC connections, which is a network layer metric, (2) Average RSRP, which is a physical layer metric.

From the figure, we can observe that physical layer attributes such as frequency and Market (topology) have a greater importance in characterizing the behavior of Average RSRP. In contrast, hardware-related attributes such

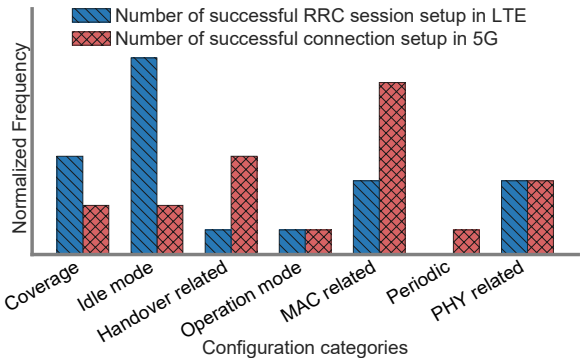


Fig. 9. The frequency of configuration categories in the set of high-impact configurations for the number of connections in LTE and 5G.

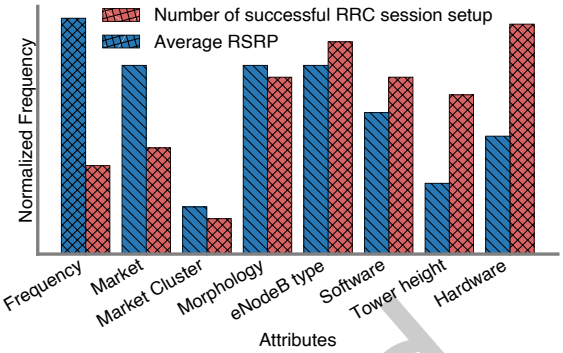


Fig. 10. Number of times each attribute appears in the top-3 contributors for characterizing the impact direction of each configuration-MAT metric pair.

as tower height and hardware versions have more impact on characterizing the behavior of the number of RRC sessions. The attributes such as morphology and eNodeB types impact both parameters mainly because they simultaneously specify the user and traffic dynamics and the physical environment around them.

9 Limitations and concluding discussions

In this work, we have demonstrated that a one-stage model directly linking the configurations to KPIs is inadequate due to the factors external of the cellular network. Consequently, we used MAT metrics as mediators to design a latent-resilient tool CIPAT to predict the direction of impact on the KPIs due to configuration changes. In this section, we highlight key limitations of CIPAT and suggest potential directions for future work.

Improving the latent resilience of CIPAT: While we establish that CIPAT is more reliable and latent-resilient solution for predicting the impact on KPI, we cannot claim that CIPAT is fully latent-invariant. Latents are by definition unknown, hence, latent-invariance of any data-driven method cannot be proved. We, however, believe that MAT metrics are the way to go for predicting the performance of the cellular network. Therefore, it is important to explore the choice of MAT metrics. Current choice of MAT metrics may not be exhaustive and additional network metrics and possibly the derived metrics can be used as MAT metrics for improved latent-resilience.

CIPAT on heterogeneous markets: While CIPAT has good performance in many markets, there are markets with very widely varying geographies and morphologies where we have insufficient amount of data samples for each attribute value to train an effective model. Our future work includes designing CIPAT for a homogeneous split of such heterogeneous markets, and potentially extrapolating results from similar attribute values across markets.

More descriptive latent-invariant characteristic: In this study, we are unable to predict the magnitude of change in KPI. This leads to a fundamental question: Is there a latent-invariant characteristic in the relationship between configuration parameters and MAT metrics that provides more information – say, a range of MATs – instead of solely the direction of change in MATs? This question is particularly intriguing considering that the proposed toolchain uses information from the first stage to support the second stage. Thus, finding a latent-invariant map between configurations to MAT metrics that is specifically designed to support the regression model in the second stage poses a difficult yet fascinating challenge.

Extension to new attributes: The use of CIPAT is limited to the specific attributes encountered during training. Both stages, the configuration to MAT and the MAT to KPI relationships, are developed for the specific attribute values. While CIPAT performs well within this domain, extrapolating to novel attribute combinations, such as

a new equipment or market, could lead to unreliable predictions. In such cases, CIPAT can be initialized using the data from similar attributes and can be progressively fine-tuned as data from the new attributes become available. A dedicated study is needed to explore principled methods for extrapolating logs to new attributes and to characterize the transferability of CIPAT, including how performance scales with the number of observed configuration-change events in a new environment.

Identifying configurations using configuration recommendation systems: CIPAT only predicts the impact of the configuration change but does not search for the best configurations. This gap can be addressed by integrating CIPAT as a discriminator in the existing configuration recommendation systems, such as Chroma [16], Aurora [25] and Auric [28]. A key limitation of these existing recommendation systems, however, is that they typically recommend only configuration settings already tested on the live network. Since CIPAT can predict the impact of untested configuration changes, we believe CIPAT could also serve as a discriminator in novel recommendation systems capable of identifying entirely new, impactful configurations.

Acknowledgments

This work is supported by NSF grants CNS-2107037 and CNS-2112471. Changhan Ge sincerely appreciates the support from Prof. Lili Qiu at UT Austin.

References

[1] Mehdi Amirijoo, Ljupco Jorguseski, Thomas Kürner, Remco Litjens, Michaela Neuland, Lars Christoph Schmelz, and Ulrich Turke. 2009. Cell outage management in LTE networks. In *6th International Symposium on Wireless Communication Systems* (Siena, Italy). IEEE, 600–604.

[2] Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. 2019. Invariant Risk Minimization. *arXiv:1907.02893* (2019).

[3] Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein generative adversarial networks. In *34th International Conference on Machine Learning* (Sydney, Australia), Vol. 70. PMLR, 214–223.

[4] Mona Azadkia and Sourav Chatterjee. 2021. A simple measure of conditional dependence. *Annals of Statistics* 49 (2021), 3070 – 3102.

[5] Ryan Beckett, Aarti Gupta, Ratul Mahajan, and David Walker. 2017. A General Approach to Network Configuration Verification. In *ACM SIGCOMM 2017* (Los Angeles, CA, USA). Association for Computing Machinery, 155–168.

[6] Emmanuel Candès, Yingying Fan, Lucas Janson, and Jinchi Lv. 2018. Panning for gold: 'model-X' knockoffs for high dimensional controlled variable selection. *J. Royal Statistical Society (Statistical Methodology)* 80, 3 (2018), pp. 551–577.

[7] Sourav Chatterjee. 2021. A new coefficient of correlation. *J. American Statistical Association* 116, 536 (2021), 2009–2022.

[8] Marco Cuturi and Arnaud Doucet. 2014. Fast computation of Wasserstein barycenters. In *31st International Conference on Machine Learning* (Bejing, China), Vol. 32. PMLR, 685–693.

[9] Ahmed El-Hassany, Petar Tsankov, Laurent Vanbever, and Martin Vechev. 2018. NetComplete: Practical Network-Wide Configuration Synthesis with Autocompletion. In *15th USENIX Symposium on Networked Systems Design and Implementation* (Renton, WA). USENIX Association, 579–594.

[10] Robin Evans and Thomas Richardson. 2014. Markovian Acyclic Directed Mixed Graphs for Discrete Data. *Annals of Statistics* 42, 4 (2014), 1452–1482.

[11] Seyed K. Fayaz, Tushar Sharma, Ari Fogel, Ratul Mahajan, Todd Millstein, Vyas Sekar, and George Varghese. 2016. Efficient Network Reachability Analysis Using a Succinct Control Plane Representation. In *12th USENIX Symposium on Operating Systems Design and Implementation* (Savannah, GA). USENIX Association, 217–232.

[12] Claudio Fiandrino, Leonardo Bonati, Salvatore D'Oro, Michele Polese, Tommaso Melodia, and Joerg Widmer. 2023. EXPLORA: AI/ML EXPLainability for the Open RAN. *Proc. of ACM on Networking* 1, 3, Article 19 (2023), 26 pages.

[13] Ari Fogel, Stanley Fung, Luis Pedrosa, Meg Walraed-Sullivan, Ramesh Govindan, Ratul Mahajan, and Todd Millstein. 2015. A General Approach to Network Configuration Analysis. In *12th USENIX Symposium on Networked Systems Design and Implementation* (Oakland, CA). USENIX Association, 469–483.

[14] Charlie Frogner, Chiyuan Zhang, Hossein Mobahi, Mauricio Araya-Polo, and Tomaso Poggio. 2015. In *29th International Conference on Neural Information Processing Systems* (Montreal, Canada), Vol. 2. MIT Press, 2053–2061.

[15] Yu Gan, Mingyu Liang, Sundar Dev, David Lo, and Christina Delimitrou. 2021. Sage: Practical and Scalable ML-Driven Performance Debugging in Microservices. In *26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems* (Virtual, USA). Association for Computing Machinery, 135–151.

[16] Changhan Ge, Zihui Ge, Xuan Liu, Ajay Mahimkar, Yusef Shaqalle, Yu Xiang, and Shomik Pathak. 2023. Chroma: Learning and Using Network Contexts to Reinforce Performance Improving Configurations. In *29th Annual International Conference on Mobile Computing and Networking* (Madrid, Spain). Association for Computing Machinery, Article 42, 16 pages.
[17] Aaron Gember-Jacobson, Raajay Viswanathan, Aditya Akella, and Ratul Mahajan. 2016. Fast Control Plane Analysis Using an Abstract Representation. In *ACM SIGCOMM 2016* (Florianopolis, Brazil). Association for Computing Machinery, 300–313.
[18] A. Ghosh, J. Zhang, J.G. Andrews, and R. Muhamed. 2010. *Fundamentals of LTE.* Pearson Education.
[19] Azam Ikram, Sarthak Chakraborty, Subrata Mitra, Shiv Kumar Saini, Saurabh Bagchi, and Murat Kocaoglu. 2022. Root cause analysis of failures in microservices through causal discovery. In *36th International Conference on Neural Information Processing Systems* (New Orleans, LA, USA). Curran Associates Inc., Article 2259, 13 pages.
[20] Amin Jaber, Jiji Zhang, and Elias Bareinboim. 2019. Causal Identification under Markov Equivalence: Completeness Results. In *36th International Conference on Machine Learning*, Vol. 97. PMLR, 2981–2989.
[21] Julie Josse and Susan Holmes. 2016. Measuring multivariate association and beyond. *Statistics surveys* 10 (2016), 132.
[22] Siva Kesava Reddy Kakarla, Alan Tang, Ryan Beckett, Karthick Jayaraman, Todd Millstein, Yuval Tamir, and George Varghese. 2020. Finding Network Misconfigurations by Automatic Template Inference. In *17th USENIX Symposium on Networked Systems Design and Implementation* (Santa Clara, CA). USENIX Association, 999–1013.
[23] Rahul Krishna, Md Shahriar Iqbal, Mohammad Ali Javidian, Baishakhi Ray, and Pooyan Jamshidi. 2020. CADET: Debugging and Fixing Misconfigurations using Counterfactual Reasoning. *arXiv:2010.06061* (2020).
[24] Mingjie Li, Zeyan Li, Kanglin Yin, Xiaohui Nie, Wenchi Zhang, Kaixin Sui, and Dan Pei. 2022. Causal Inference-Based Root Cause Analysis for Online Service Systems with Intervention Recognition. In *28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (Washington DC, USA). Association for Computing Machinery, 3230–3240.
[25] Ajay Mahimkar, Zihui Ge, Xuan Liu, Yusef Shaqalle, Yu Xiang, Jennifer Yates, Shomik Pathak, and Rick Reichel. 2022. Aurora: conformity-based configuration recommendation to improve LTE/5G service. In *22nd ACM Internet Measurement Conference* (Nice, France). Association for Computing Machinery, 83–97.
[26] Ajay Mahimkar, Zihui Ge, Jia Wang, Jennifer Yates, Yin Zhang, Joanne Emmons, Brian Huntley, and Mark Stockert. 2011. Rapid Detection of Maintenance Induced Changes in Service Performance. In *7th Conference on Emerging Networking Experiments and Technologies* (Tokyo, Japan). Association for Computing Machinery, Article 13, 12 pages.
[27] Ajay Mahimkar, Zihui Ge, Jennifer Yates, Chris Hristov, Vincent Cordaro, Shane Smith, Jing Xu, and Mark Stockert. 2013. Robust Assessment of Changes in Cellular Networks. In *9th ACM Conference on Emerging Networking Experiments and Technologies* (Santa Barbara, USA). Association for Computing Machinery, 175–186.
[28] Ajay Mahimkar, Ashiwan Sivakumar, Zihui Ge, Shomik Pathak, and Karunasish Biswas. 2021. Auric: Using Data-driven Recommendation to Automatically Generate Cellular Configuration. In *ACM SIGCOMM 2021* (Virtual, USA). Association for Computing Machinery, 807–820.
[29] Ajay Mahimkar, Han Hee Song, Zihui Ge, Aman Shaikh, Jia Wang, Jennifer Yates, Yin Zhang, and Joanne Emmons. 2010. Detecting the Performance Impact of Upgrades in Large Operational Networks. In *ACM SIGCOMM 2010* (New Delhi, India). Association for Computing Machinery, 303–314.
[30] Boris Muzellec, Julie Josse, Claire Boyer, and Marco Cuturi. 2020. Missing Data Imputation using Optimal Transport. In *37th International Conference on Machine Learning*, Vol. 119. PMLR, 7130–7140.
[31] Kartik Patel, Changhan Ge, Ajay Mahimkar, Sanjay Shakkottai, and Yusef Shaqalle. 2024. CIPAT: Latent-resilient Toolkit for Performance Impact Prediction due to Configuration Tuning. In *30th Annual International Conference on Mobile Computing and Networking* (Washington D.C., USA). Association for Computing Machinery, 2377–2382.
[32] Kartik Patel, Changhan Ge, Ajay Mahimkar, Sanjay Shakkottai, and Yusef Shaqalle. 2024. Predicting the Performance of Cellular Networks: A Latent-resilient Approach. In *30th Annual International Conference on Mobile Computing and Networking* (Washington D.C., USA). Association for Computing Machinery, 1581–1583.
[33] Judea Pearl. 2009. *Causality.* Cambridge University Press.
[34] Emilija Perkovic, Johannes Textor, Markus Kalisch, and Marloes H. Maathuis. 2017. Complete Graphical Characterization and Construction of Adjustment Sets in Markov Equivalence Classes of Ancestral Graphs. *Journal of Machine Learning Research* 18, 1 (Jan. 2017), 8132–8193.
[35] Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. 2017. *Elements of Causal Inference: Foundations and Learning Algorithms.* MIT Press.
[36] Santhosh Prabhu, Kuan Yen Chou, Ali Kheradmand, Brighten Godfrey, and Matthew Caesar. 2020. Plankton: Scalable Network Configuration Verification through Model Checking. In *17th USENIX Symposium on Networked Systems Design and Implementation* (Santa Clara, CA). USENIX Association, 953–967.
[37] Juan Qiu, Qingfeng Du, Kanglin Yin, Shuang-Li Zhang, and Chongshu Qian. 2020. A Causality Mining and Knowledge Graph Based Method of Root Cause Diagnosis for Performance Anomaly in Cloud Applications. *Applied Sciences* 10, 6 (2020).
[38] Thomas Richardson and Peter Spirtes. 2002. Ancestral Graph Markov Models. *Annals of Statistics* 30, 4 (2002), 962–1030.

[39] Nihal Sharma, Soumya Basu, Karthikeyan Shanmugam, and Sanjay Shakkottai. 2020. On Under-exploration in Bandits with Mean Bounds from Confounded Data. *arXiv:2002.08405* (2020).

[40] Joel Shodamola, Usama Masood, Marvin Manalastas, and Ali Imran. 2020. A Machine Learning-based Framework for KPI Maximization in Emerging Networks using Mobility Parameters. In *IEEE International Black Sea Conference on Communications and Networking* (Odessa, Ukraine). 1–6.

[41] Ricardo Silva and Zoubin Ghahramani. 2009. The Hidden Life of Latent Variables: Bayesian Learning with Mixed Graph Models. *Journal of Machine Learning Research* 10, 41 (2009), 1187–1238.

[42] Samuel Steffen, Timon Gehr, Petar Tsankov, Laurent Vanbever, and Martin Vechev. 2020. Probabilistic Verification of Network Configurations. In *ACM SIGCOMM 2020* (Virtual, USA). Association for Computing Machinery, 750–764.

[43] Xudong Sun, Runxiang Cheng, Jianyan Chen, Elaine Ang, Owolabi Legunsen, and Tianyin Xu. 2020. Testing Configuration Changes in Context to Prevent Production Failures. In *14th USENIX Symposium on Operating Systems Design and Implementation*. USENIX Association, 735–751.

[44] Bingchuan Tian, Xinyi Zhang, Ennan Zhai, Hongqiang Harry Liu, Qiaobo Ye, Chunsheng Wang, Xin Wu, Zhiming Ji, Yihong Sang, Ming Zhang, et al. 2019. Safely and automatically updating in-network ACL configurations with intent language. In *ACM SIGCOMM 2019*. Association for Computing Machinery, 214–226.

[45] Ping Wang, Jingmin Xu, Meng Ma, Weilan Lin, Disheng Pan, Yuan Wang, and Pengfei Chen. 2018. CloudRanger: Root Cause Identification for Cloud Native Systems. In *18th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*. 492–502.

[46] Li Wenjing, Yu Peng, Jiang Zhengxin, and Li Zifan. 2012. Centralized Management Mechanism for Cell Outage Compensation in LTE Networks. *International Journal of Distributed Sensor Networks* 8, 11 (2012), 170589.

[47] Xing Xu, Ioannis Broustis, Zihui Ge, Ramesh Govindan, Ajay Mahimkar, N. K. Shankaranarayanan, and Jia Wang. 2015. Magus: Minimizing Cellular Service Disruption during Network Upgrades. In *11th ACM Conference on Emerging Networking Experiments and Technologies* (Heidelberg, Germany). Association for Computing Machinery, Article 21, 13 pages.

[48] Steve Yadlowsky, Hongseok Namkoong, Sanjay Basu, John Duchi, and Lu Tian. 2022. Bounds on the Conditional and Average Treatment Effect with Unobserved Confounding Factors. *Annals of Statistics* 50, 5 (2022), 2587–2615.

[49] Kun Zhang, Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. 2011. Kernel-Based Conditional Independence Test and Application in Causal Discovery. In *27th Conference on Uncertainty in Artificial Intelligence* (Barcelona, Spain). AUAI Press, Arlington, Virginia, USA, 804–813.

A Sample list of MAT metrics

The sample list is presented in Table 7.

Category	MAT Metric	MAT Metric
Mobility	Number of Inter-frequency HO entered in execution phase	Number of intra-frequency HO entered in preparation phase
	Number of Inter-frequency HO with successful execution phase	Number of intra-frequency HO with successful preparation phase
	Success rate of inter-frequency HO in execution phase	Success rate of intra-frequency HO in preparation phase
	Number of inter-frequency HO entered in preparation phase	Number of intra-frequency HO entered in execution phase
	Number of inter-frequency HO with successful preparation phase	Number of intra-frequency HO with successful execution phase
	Success rate of inter-frequency HO in preparation phase	Success rate of intra-frequency HO in execution phase
	Number of attempted inter-frequency HO	Mode of distance b/w UEs and the BS (km)
	Number of successful inter-frequency HO	10%-tile distance between UEs and the cell
	Success rate of inter-frequency HO	25%-tile distance between UEs and the cell
	Number of attempted intra-frequency HO	50%-tile distance between UEs and the cell
	Number of successful intra-frequency HO	75%-tile distance between UEs and the cell
	Success rate of intra-frequency HO	85%-tile distance between UEs and the cell
Access	Average RSRP in dBm	# of UEs with -14 <RSRQ <-10 dB
	# of UEs with RSRP >-90 dBm	# of UEs with -18 <RSRQ <-14 dB
	# of UEs with -100 <RSRP <-90 dBm	# of UEs with RSRQ <-18
	# of UEs with -110 <RSRP <-100 dBm	Average channel quality indicator (CQI)
	# of UEs with -115 <RSRP <-110 dBm	Average signal-to-noise-ratio (SNR)
	# of UEs with -118 <RSRP <-115 dBm	# of UEs with RSRP >-106 dBm and RSRQ <-16 dB
	# of UEs with RSRP <-118 dBm	# of UEs with RSRP >-110 dBm and RSRQ <-14 dB
	Average RSRQ in dB	# of UEs with RSRP >-112 dBm and RSRQ <-18 dB
	# of UEs with RSRQ >-10 dB	# of UEs with -118<RSRP<-112 dBm and RSRQ <-18 dB
Traffic	Number of attempted RRC connection	Number of attempted S1MME connection
	Number of established RRC connection	Number of established S1MME connection
	RRC establishment success rate	S1MME establishment success rate
	Downlink PDCP volume	Total number of PDCCH resources
	Downlink packet loss	Number of PDCCH resources used
	Number of DRB established	PDCCH Utilization
	Number of downlink physical resource blocks	Number of uplink physical resource blocks

Table 7. A sample list of MAT metrics: The parameters in blue are derived from the observed parameters (usually ratios). These parameters are not considered in the first stage of CIPAT as they do not maintain the same direction of change *w.r.t.* configuration changes. However, we use them as an input to the regression model in the second stage to reduce the burden on the model to learn the derived parameters. For detailed explanation on HO procedure and the impact of configurations on them, please refer to [18].

Received 22 February 2025; revised 2 January 2026; accepted 21 February 2026