Multi-Task Combinatorial Bandits for Budget Allocation

Lin Ge
North Carolina State University
Raleigh, NC, USA

Yang Xu
North Carolina State University
Raleigh, NC, USA

Jianing Chu
North Carolina State University
Raleigh, NC, USA

David Cramer
Amazon
Seattle, WA, USA

Fuhong Li
Amazon
Seattle, WA, USA

Kelly Paulson
Amazon
Seattle, WA, USA

Rui Song
Amazon
Seattle, WA, USA

ABSTRACT

Today's top advertisers typically manage hundreds of campaigns simultaneously and consistently launch new ones throughout the year. A crucial challenge for marketing managers is determining the optimal allocation of limited budgets across various ad lines in each campaign to maximize cumulative returns, especially given the huge uncertainty in return outcomes. In this paper, we propose to formulate budget allocation as a multi-task combinatorial bandit problem and introduce a novel online budget allocation system. The proposed system: i) integrates a Bayesian hierarchical model to intelligently utilize the metadata of campaigns and ad lines and budget size, ensuring efficient information sharing; ii) provides the flexibility to incorporate diverse modeling techniques such as Linear Regression, Gaussian Processes, and Neural Networks, catering to diverse environmental complexities; and iii) employs the Thompson sampling (TS) technique to strike a balance between exploration and exploitation. Through offline evaluation and online experiments, our system demonstrates robustness and adaptability, effectively maximizing the overall cumulative returns. A Python implementation of the proposed procedure is available at https://anonymous.4open.science/r/MCMAB.

KEYWORDS

Online Advertising, Budget Allocation, Combinatorial Bandits, Meta Bandits

ACM Reference Format:
Lin Ge, Yang Xu, Jianing Chu, David Cramer, Fuhong Li, Kelly Paulson, and Rui Song. 2018. Multi-Task Combinatorial Bandits for Budget Allocation. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation emai (Conference acronym 'XX).* ACM, New York, NY, USA, 6 pages. https://doi.org/XXXXXXX.XXXXXXX

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY

ACM ISBN 978-1-4503-XXXX-X/18/06
https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Budget allocation has been given wide attention in the advertising market [1, 9]. Advertisers and agencies that use Demand Side Platforms (DSP), like Amazon DSP, routinely manage hundreds of simultaneous campaigns, each comprising various ad lines targeting specific audiences and set with diverse delivery settings. Daily, marketing managers allocate budgets across ad lines for each campaign within a daily budget, aiming to boost traffic to retail websites or maximize product sales. A key challenge is the lack of understanding of the relationship between ads spending and performance outcomes, which, once obtained, reduces the task to an optimization problem solvable by various methods [2, 7].

Today's ADSP implements automated online performance optimizations that respond to signals on each campaign's own spend and conversion data. Such an approach often encounters two significant challenges: **First**, advertisers frequently launch campaigns sequentially or run multiple campaigns simultaneously. Without a design in the ADSP that effectively coordinates learning from past and concurrent campaigns, the learning process is inefficient. Ad lines typically begin with an equal budget distribution at the start of a campaign. Although automated tools can make real-time budget adjustments during campaigns, determining the optimal budget allocation mix can take up to three weeks. This delay would result in a considerable number of ineffective ad deliveries, a general issue particularly severe for short-lived campaigns and large advertisers or agencies participating in numerous campaigns each year. **Second**, measurement uncertainty exists, primarily due to the lack of real-life counterfactual observations, necessitating the use of estimations. This challenge is exacerbated by the dynamic nature of advertising and business environments, where factors like seasonality, competitive bidding, privacy regulations, DSP functionality, and fluctuating customer behaviors make it even more complex to accurately estimate these values. Relying solely on historical data can lead to suboptimal solutions in sequential decision-making scenarios. Thus, the following question is addressed in this paper: ***How can we wisely utilize i) learnings from past campaigns and ii) insights from other ongoing concurrent campaigns to accelerate the learning process for optimal budget allocation?***

To address the inherent uncertainty in digital advertising, Bandits algorithms are known for effectively balancing exploration and exploitation and have recently been applied to budget allocation, formalizing it as a combinatorial multi-armed bandit (CMAB)

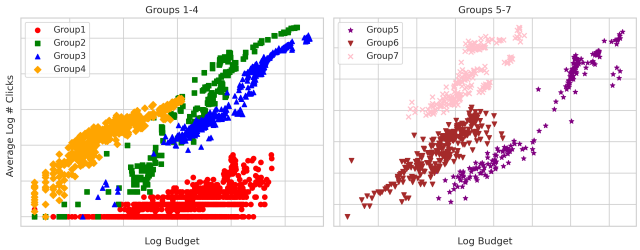

Figure 1: Scatter plots of the log of average number of clicks received and the log of budget allocated for various *ad line groups* with distinct advertisers' industries, channels, supply sources, and audiences.

problem [10, 11, 20]. However, none of them investigated how to share information across multiple ad lines and campaigns. Figure 1 depicts the average clicks received for different ad lines from various campaigns, with varying levels of budget assigned, underscoring the significant influence of ad/campaign characteristics on budget-performance relationships. Additionally, according to Figure 1, it is important to note that the available features do not fully determine the expected performance of an ad line with an assigned specific budget. In other words, even when conditioned on informative features, the expected performance for a given budget level still exhibits a degree of variability (referred to as *inter-arm heterogeneity*). To strategically boost information sharing across tasks, we model budget allocation in ADSP as a multi-task CMAB problem where each ad campaign represents a single CMAB task, and introduce a feature-based Bandit algorithm augmented by a Bayesian hierarchical model.

Contributions. Our main contributions are multi-fold.

First, our study is the first, to our knowledge, to investigate budget allocation for online advertising from the perspective of large advertisers and agencies managing multiple campaigns, with a meta objective of optimizing performance over the campaign distribution. This contrasts with existing studies that primarily focus on optimizing a single campaign from an advertiser's viewpoint [4, 10, 11].

Second, to capture the budget-performance relationship, we propose a general Bayesian hierarchical model that supports both parametric and non-parametric modeling. Driven by similar motivations, Han and Arndt [4] introduced a contextual bandit-based system using augmented data generated from a global model to share information across ad lines. However, not accounting for the uncertainty of the fitted global model, its effectiveness relies heavily on the global model's performance and can lead to suboptimal decisions, especially in data-limited scenarios. Furthermore, with the power law assumption, they assume a linear relationship between the logarithms of the budget and the performance metric, which often fails in practice, as shown in Figure 1. In contrast, our method can effectively address more complex nonlinear budget-performance relationships by integrating with the Gaussian Process and Neural Network, as two instances.

Third, through the construction of the Bayesian hierarchical model, which incorporates an arm-specific random effect to capture the information not being explained by the feature information, we effectively tackle the inter-arm heterogeneity observed in Figure 1. While none of the aforementioned work addresses this ubiquitous issue, recent advancements in meta bandits [15, 16] likewise focus on dealing with such heterogeneity. However, they all rely on parametric linear model assumptions.

Finally, we implemented our proposed framework in an offline study using real campaign data from ADSP. The results consistently indicate that our framework achieves faster convergence and higher cumulative reward, thereby leading to a better budget allocation strategy in the long term. This is also supported by an online experiment.

2 RELATED WORK

Budget/resource allocation, extensively explored over the past decades, has recently been formalized within the framework of CMAB. Formulating the allocation problem as CMAB [3, 10, 11, 17, 18, 20], budgets are discretized into finite proportions, aligning with the nature of combinatorial bandits in slate recommendation. Notably, Zuo and Joe-Wong [20] leveraged the CMAB framework by defining a super arm as an (ad line, budget) tuple for action assignment, and naturally extended this idea to continuous budget allocation scenarios with additional Lipschitz continuity assumptions. In the work by Xu et al. [18], the authors studied resource allocation problem with concave objective and fairness constraints. Nuara et al. [10, 11] proposed a joint bid/budget optimization algorithm based on Bayesian bandits update with Gaussian process. In the work by Gupta et al. [3], the authors proposed a correlated combinatorial bandit framework to capture the structural correlations between reward functions. However, these methods either fail to utilize contextual information [17, 20] or rely on restrictive modeling assumptions in rewards and resource consumptions [18], limiting their applicability to more general applications.

To utilize the contextual information expediting the learning process, Han and Gabor [5] introduced a contextual bandit framework via a global-local model. However, Han and Gabor [5] focuses on a single budget allocation task and ignores the inter-arm heterogeneity. Additionally, their updating procedure lacks effective exploration of the global model, making the methodology's performance heavily dependent on how well the global model fits the data. This poses a potential challenge when dealing with limited sample sizes or significant noise in the data. In this work, we build upon the combinatorial bandit framework to do information sharing. While discretizing budgets in CMAB may introduce a minor bias in the precision of estimating the optimal arm, this approach eliminates the necessity of imposing smoothness assumptions, which is typically required when the budget is considered continuous [5]. Additionally, our work is closely related to traditional approaches that consider budget allocation purely as an optimization problem without addressing estimation uncertainty Ou et al. [12], and the body of work on Multi-task/Meta Bandits [15].

3 PRELIMINARIES

3.1 Budget Allocation

Consider a large advertiser hosting a collection of " advertising campaigns running either concurrently or sequentially. In each campaign < 2 »"¼ = f1•2•"""•"g, there are < ad lines designated for daily budget allocation, and the campaign duration is

denoted as $)_{<}$. For a given campaign $<$, we assume a total daily budget of $_{<}$. At each time round $\mathbb{C}$, the budget assigned to each ad line $:$ 2 » $_{<}¼$ is denoted as $0_{<\cdot:\cdot\mathbb{C}}$ and a constraint exists such that the sum of the assigned budgets across all ad lines in campaign $<$ satisfies $\sum_{:=1}^{<} 0_{<\cdot:\cdot\mathbb{C}}$ $_{<}$. After assigning budget $0_{<\cdot:\cdot\mathbb{C}}$ for each ad line $:$ in campaign $<$, we consequently observe a random reward $'_{<\cdot:\cdot\mathbb{C}}{}^{1}0_{<\cdot:\cdot\mathbb{C}}{}^{o}$. Our goal is to maximize the cumulative reward function across all ad lines, all campaigns and all rounds:

$$\begin{aligned} \underset{0_{<\cdot:\cdot\mathbb{C}}}{\text{maximize}} \quad & \sum_{<=1}^{} \sum_{\mathbb{C}=1}^{} \sum_{:=1}^{} \mathsf{Ef}\,'_{<\cdot:\cdot\mathbb{C}}{}^{1}0_{<\cdot:\cdot\mathbb{C}}{}^{o}g\cdot \\ \text{subject to} \quad & \sum_{:=1}^{} 0_{<\cdot:\cdot\mathbb{C}} \quad _{<} \quad 8<\cdot\mathbb{C}'' \end{aligned} \tag{1}$$

3.2 Combinatorial Multi-Armed Bandits

To connect the optimization problem described above with the framework of CMAB, we begin by introducing the fundamental concept of CMAB. A typical CMAB problem consists of arms associated with a set of random variables $'_{:\cdot\mathbb{C}}$. The random variable $'_{:\cdot\mathbb{C}}$ indicates the random reward of arm $:$ 2 » ¼ at time round $\mathbb{C}$. The set of all possible subsets of arms is a power set $\mathsf{S} = 2^{»}$ ¼. We refer to every set of arms $($ 2 S as a super arm and every arm in $($ as a base arm. At each time round, one super arm $($ 2 S is played and the rewards of all base arms in this super arm are observed.

We consider each campaign as a single-task CMAB problem where each ad line within the campaign corresponds to a base arm. The first challenge is that, unlike the conventional CMAB problem where the decision revolves around whether to play the base arm or not, in our scenario, we must also determine the budget allocation for each chosen base arm. As the budget amount is continuous, we confront an infinite number of potential base arms. The second challenge arises from the presence of daily budget constraints. This implies that, at each round, only a subset of super arms is viable for play, restricted by the limitations imposed by the available daily budget. Even upon overcoming these challenges, conventional CMAB are designed to accommodate only a single campaign. However, advertisers often initiate new campaigns or run multiple campaigns concurrently. To mitigate the cold start issue associated with new campaigns and enhance data utilization, the proposed CMAB must facilitate information sharing across different campaigns.

4 METHODOLOGY

4.1 Problem Formulation

Without loss of generality, we define the continuous action space as $\mathsf{A} = »0\cdot 1¼$, where $0_{<\cdot:\cdot\mathbb{C}}$ 2 A represents the proportion of the total budget allocated to ad line $:$ in campaign $<$ at time round $\mathbb{C}$. The corresponding budget can be expressed as $_{<}$ $0_{<\cdot:\cdot\mathbb{C}}$. We further discretize the action space by partitioning the continuous budget into different proportions: $\mathsf{A}_{3} = \mathsf{f}0\cdot \frac{1}{\#}\cdot \frac{2}{\#}\cdot \cdots \cdot \frac{\#-1}{\#}\cdot 1\mathsf{g}$, with # denoting a user-specified integer constant. The rationale behind the discretization is twofold. First, in practical scenarios, campaign budgets are commonly assigned in rounded percentages, such as 10% or 25%, rather than extremely precise amounts. Second, discretization eliminates the need for smoothness assumptions typically required for continuous budget optimization. Accordingly,

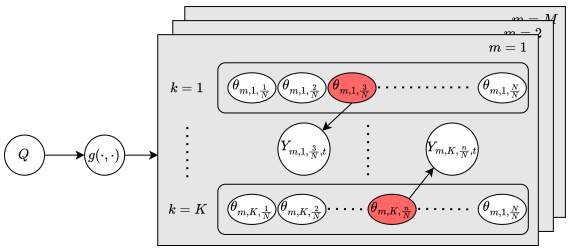

Figure 2: Graphical representation of model (3). Red nodes are the selected base arm at round $\mathbb{C}$.

we consider maintaining a set of base arms in campaign $<$ as $\mathsf{f}^{1}:\cdot 0^{o}$ j $:$ 2 » $_{<}¼\cdot 0$ 2 $\mathsf{A}_{3}\mathsf{g}$. Each base arm contains metadata $\mathsf{G}_{<\cdot:}$,which comprises specific information on campaign and ad line configurations. At each time round $\mathbb{C}$, we can only play one arm $^{1}:\cdot 0^{o}$ for each ad line $:$ in campaign $<$. This implies the allocation of a budget proportion $0_{<\cdot:\cdot\mathbb{C}}$ to ad line $:$ in campaign $<$. Let $\backslash_{<\cdot:\cdot 0}$ $\mathsf{Ef}\,'_{<\cdot:}{}^{1}0^{o}\mathsf{g}$. We can rewrite our goal as:

$$\begin{aligned} \underset{0_{<\cdot:\cdot\mathbb{C}}}{\text{maximize}} \quad & \sum_{<=1}^{} \sum_{\mathbb{C}=1}^{} \sum_{:=1}^{} \backslash_{<\cdot:\cdot 0}\, 0_{<\cdot:\cdot\mathbb{C}}\cdot \\ \text{subject to} \quad & \sum_{:=1}^{} 0_{<\cdot:\cdot\mathbb{C}} \quad 1\cdot \sum_{==0}^{\#} \quad 0_{<\cdot:\cdot\mathbb{C}} = \frac{=}{\#} \quad = 1\cdot \quad 8<\cdot\mathbb{C}'' \end{aligned} \tag{2}$$

Thus, let $\boldsymbol{a}_{<\mathbb{C}} = {}^{1}0_{<\cdot 1\cdot\mathbb{C}}\cdot \quad \cdot 0_{<\cdot \quad <\mathbb{C}}{}^{o}$, the full allocation space for each campaign $<$ at each decision point is $\mathsf{S}_{<} = \mathsf{f}\boldsymbol{a}_{<\cdot}$ j $0_{<\cdot:\cdot}$ 2 $\mathsf{A}_{3}\cdot$ $\backslash_{:=1}^{<} 0_{<\cdot:\cdot} \quad 1\cdot \backslash_{==0}^{\#} \quad 0_{<\cdot:\cdot} = \frac{=}{\#} = 1\cdot 8:\mathsf{g}$.

4.2 Multi-task Bayesian Hierarchical CMAB Framework

To tackle the optimization problem (2), a key step is the estimation of $\backslash_{<\cdot:\cdot 0}$, which we propose to accomplish using the following Bayesian hierarchical model:

$$\begin{aligned} &\text{(Prior)} && \text{Prior information Q related to } 6\cdot \\ &\text{(Generalization)} && \backslash_{<\cdot:\cdot 0}\,\mathsf{j}\,\mathsf{x}_{<\cdot:\cdot 0} = 6^{1}\mathsf{x}_{<\cdot:\cdot 0}{}^{\cdot 0^{o}}{}_{,}\, X_{<\cdot:\cdot 0}\cdot \\ &&& 8< \; 2 \; »''¼\cdot: \; 2 \; » \; _{<}¼\cdot 0 \; 2 \; \mathsf{A}_{3}\cdot \\ &\text{(Observation)} && \cdot_{<\cdot:\cdot 0_{<\cdot:\cdot\mathbb{C}}\mathbb{C}} = \backslash_{<\cdot:\cdot 0_{<\cdot:\cdot\mathbb{C}}}{}_{,}\, n_{<\cdot:\cdot\mathbb{C}}\cdot \\ &\text{(Reward)} && '_{<\mathbb{C}} = \sum_{:\,2\,»\;<¼}^{} \cdot_{<\cdot:\cdot 0_{<\cdot:\cdot\mathbb{C}}\mathbb{C}}\cdot \end{aligned} \tag{3}$$

where $\backslash_{<\cdot:\cdot 0}$ is the expected reward of allocating a budget of $_{<}$ 0 to ad line $:$ in campaign $<$ and $n_{<\cdot:\cdot\mathbb{C}}$ $\#^{1}0\cdot f_{n}^{2o}$ is the random noise for some known f_{n}. At round $\mathbb{C}$, $\cdot_{<\cdot:\cdot 0\mathbb{C}}$ is the observed reward for ad line $:$ and $'_{<\mathbb{C}}$ is the total reward aggregating the observed rewards from all ad lines within campaign $<$. The essence of (3) lies in the two-way decomposition of $\backslash_{<\cdot:\cdot 0}$, which splits $\backslash_{<\cdot:\cdot 0}$ into two components: i) $6^{1}\mathsf{x}_{<\cdot:}{}^{\cdot 0^{o}}$, a function capturing the average impact of available features $\mathsf{x}_{<\cdot:}$ and action 0 on the reward, with $\&$ as the prior belief about 6's distribution, and ii) $X_{<\cdot:\cdot 0}$, a random effect that accounts for the inter-arm heterogeneity conditioned on $\mathsf{x}_{<\cdot:}$ and 0. See Figure 2 for an illustration. Recognizing that even the most advanced machine learning algorithms cannot perfectly represent the relationship between features and reward, the additional random effect is primarily employed to account for the

uncertainty in $\mathbf{l}_{<,:,0}$ that failed to be captured by θ. Intuitively, as such, we utilize i) the shared information across campaigns and ad lines via θ and ii) the observations $\cdot_{<,:,0<,:,\mathcal{C}}$ from all correlated base arms, to infer $\mathbf{l}_{<,:,0}$.

Let $\mathbf{X}_< = X_{<,1,0}, \ldots, X_{<,\cdot,\frac{\#}{\#}}$, we assume that $\mathbf{X}_< \sim \#(0, \Sigma)$ for some known covariance matrix Σ. For θ, either a parametric or non-parametric model can be used. In this work, we consider three working models as examples: i) Linear regression (LR), assuming $\theta^{\intercal}\mathbf{x}_{<,:,0} = q(\mathbf{x}_{<,:,0})^{\intercal}\boldsymbol{\beta}$, where $q(\mathbf{x}_{<,:,0})$ is a certain transformation of features and actions, and $\boldsymbol{\beta}$ is a vector of parameters with a prior $\boldsymbol{\beta} \sim \#(\boldsymbol{\gamma}, \Sigma_{\boldsymbol{\gamma}})$. ii) Neural network (NN) regression, considering $\theta^{\intercal}\mathbf{x}_{<,:,0}$ as a fully connected NN of depth $L \geq 2$, the collection of parameters of which, $\boldsymbol{\beta}$, has a prior $\boldsymbol{\beta} \sim \#(\boldsymbol{\gamma}, \Sigma_{\boldsymbol{\gamma}})$ [6]. iii) Gaussian process (GP) regression, assuming that $\theta^{\intercal}\mathbf{x}_{<,:,0}$ follows a Gaussian process prior, such that $\theta \sim \mathrm{GP}(\grave{}_{\boldsymbol{\gamma}}, K_{\boldsymbol{\gamma}})$.

4.3 Learning Strategy

4.3.1 Posterior Distributions. To sequentially update the parameter estimation in an online setting, a key step is to derive the posterior distribution of parameters in (3). Let $\boldsymbol{\beta}_< = \{\mathbf{l}_{<,:,0}\}_{1 \leq \cdot \leq 1}$ and $\boldsymbol{\beta} = \{\boldsymbol{\beta}_<\}_{1 \leq < \leq \#}$ as a $\sum_{<\geq 2} \#_< \times \#$-dimensional vector containing the expected reward for all $(<,:,0)$ tuples. Since $\mathbb{P}(\boldsymbol{\beta} \mid \mathcal{H}) \propto \mathbb{P}(\boldsymbol{\beta} \mid \mathcal{H}, \theta)\mathbb{P}(\theta \mid \mathcal{H})$, we split the posterior derivation into two parts: 1) $\mathbb{P}(\theta \mid \mathcal{H})$, and 2) $\mathbb{P}(\boldsymbol{\beta} \mid \mathcal{H}, \theta)$.

$\mathbb{P}(\theta \mid \mathcal{H})$ shares a general structure under LR, GP and NN. Let $\boldsymbol{\Psi}_{1:}$ be a 3 matrix comprising features of the selected arms (one for each ad line) offered from round 1 to round . Similarly, $\mathbf{r}_{1:} = (r_1, \ldots, r_)$ denotes the observed rewards of all base arms offered up to round . $\mathbb{P}(\theta \mid \mathcal{H})$ follows a normal distribution, with mean and covariance as

$$\mathrm{E}(\theta \mid \mathcal{H}) = \grave{}_1, \quad K(\cdot, \boldsymbol{\Psi}_{1:})[\Phi + I_{\Sigma_{1:}}]^{-1}\mathbf{r}_{1:} \quad \grave{}_{\boldsymbol{\Psi}_{1:}}$$

$$\mathrm{Cov}(\theta \mid \mathcal{H}) = K(\cdot, \cdot) - K(\cdot, \boldsymbol{\Psi}_{1:})[\Phi + I_{\Sigma_{1:}}]^{-1}K(\boldsymbol{\Psi}_{1:}, \cdot),$$

where $\Phi = K(\boldsymbol{\Psi}_{1:}, \boldsymbol{\Psi}_{1:}) + \Sigma_{1:}$. Here, $K(\boldsymbol{\Psi}_{1:}, \boldsymbol{\Psi}_{1:}) + \Sigma_{1:}$ is the variance induced by the prior distribution, and $\Sigma_{1:}$ denotes the variance induced by the random effect. In LR, $\grave{}_{\boldsymbol{\gamma}}(G) = G\boldsymbol{\gamma}$ takes a specific linear form, and $K(G, G') = G\Sigma_{\boldsymbol{\gamma}}G'$. In GP, $\grave{}(G)$ as the prior mean can adopt any function form of G, and $K(G, G')$ is a general kernel function. It can be a linear kernel $K(G, G') = \langle G, G' \rangle$, an RBF kernel $K(G, G') = \exp\left(-\frac{\|G - G'\|^2}{2\ell^2}\right)$ with ℓ as a hyperparameter representing the standard deviation, or other kernel functions. In NN, $\grave{}(G)$ is a fully-connected neural network, and $K(G, G')$ is the neural tangent kernel.

Given θ, $\mathbb{P}(\boldsymbol{\beta}_< \mid \mathcal{H}, \theta)$ follows a normal distribution with

$$\mathrm{E}(\boldsymbol{\beta}_< \mid \mathcal{H}, \theta) = \mathrm{Cov}(\boldsymbol{\beta}_< \mid \mathcal{H}, \theta)\left(\Sigma^{-1}\boldsymbol{\gamma} + \Psi_<^{\intercal}\mathbf{r}_<\right)$$

$$\mathrm{Cov}(\boldsymbol{\beta}_< \mid \mathcal{H}, \theta) = \left(\Sigma^{-1} + \ell^{-2}\Psi_<^{\intercal}\Psi_<\right)^{-1},$$

where $\#_<$ is the number of observations in $\mathcal{H}$ that correspond to base arm $(<,:,=)$.

4.3.2 TS and Optimization. Using the derived posteriors, we adopt the classical TS-type algorithm but split the posterior sampling into two steps for each decision point. Specifically, we first sample $\theta \sim \mathbb{P}(\theta \mid \mathcal{H})$ and then sample $\tilde{\boldsymbol{\beta}}_< \sim \mathbb{P}(\boldsymbol{\beta}_< \mid \theta, \mathcal{H})$. In the first step, we integrate all collected data to create a feature-based informative prior for each $\mathbf{l}_{<,:,0}$, which then guides the subsequent learning of $\mathbf{l}_{<,:,0}$. We refer to this as a feature-guided (FG) approach. This notably differs from that of [4], which employs only the first step.

Algorithm 1: Multi-Task Combinatorial Bandits (MCMAB)

Input : Specification of θ and the corresponding prior; known parameters (i.e., f_n, Σ); $\mathcal{H} = \{\}$

for *every decision point* j **do**

- Retrieve the campaign index $<$;
- Update the posterior for θ as $\mathbb{P}(\theta \mid \mathcal{H})$, according to (3);
- Sample a $\theta \sim \mathbb{P}(\theta \mid \mathcal{H})$;
- Given θ, update the posterior for $\boldsymbol{\beta}_<$ as $\mathbb{P}(\boldsymbol{\beta}_< \mid \theta, \mathcal{H})$;
- Sample an utility vector $\tilde{\boldsymbol{\beta}}_< \sim \mathbb{P}(\boldsymbol{\beta}_< \mid \theta, \mathcal{H})$;
- Take action $\mathbf{a}_{<,\mathcal{C}} = \mathrm{OA}(\theta)_{<,\mathcal{G}}\left(\mathbf{a}_{<,\cdot} \in 2^{\mathcal{S}_<} : 2 \leq \sum_{<\leq \#} \tilde{\mathbf{l}}_{<,:,0<,:,\cdot}\right)$;
- Receive reward $\mathbf{r}_{<,\mathcal{C}}$;
- Update the dataset as $\mathcal{H} \leftarrow \mathcal{H} \cup \{(<, \mathbf{a}_{<,\mathcal{C}}, \mathbf{r}_{<,\mathcal{C}})\}$

end

Given $\tilde{\boldsymbol{\beta}}_<$, the final step involves an optimization problem. Specifically, we need to solve $\underset{\mathbf{a}_{<,\mathcal{C}} \in 2^{\mathcal{S}_<}}{\mathrm{argmax}} \left(: 2 \leq \sum_{<\leq \#} \tilde{\mathbf{l}}_{<,:,0<,:,\mathcal{C}}\right)$ which can be regarded as a Multiple-Choice Knapsack Problem (MCKP) [8]. Specifically, MCKP is a generalization of the ordinary knapsack problem, where the set of items are originally partitioned into groups. Instead of making binary choices regarding each item, MCKP only allows (at most) one item in the same group to be chosen. Similar to the ordinary knapsack problem, one can utilize dynamic programming to find the optimal solution. We summarize the entire learning strategy in Algorithm 1.

5 OFFLINE EVALUATION

In order to assess the effectiveness of the proposed approach in real-world settings, we compare it with the existing methods using the Amazon Digital Advertisements' campaign data from the first quarter of 2023.

Design. To simulate real-world scenarios, we first determined θ, $f_<$, and f_n. The selection of θ involved comparing the performance of linear regression, random forest, and CatBoost [13] in predicting the logarithm of clicks obtained (i.e., $\log(\mathbf{l}_{<,:,0})$) using campaigns/ad lines' metadata (i.e., supply source, channel) and the logarithm of budget cost. CatBoost emerged as the most accurate, exhibiting the lowest mean squared error. Subsequently, let $\Sigma = \sigma^2_{\epsilon}\mathbf{I}$, $f_<$ and f_n were determined to be 0.35 and 0.40, respectively, using maximum likelihood estimation based on the fitted CatBoost model and under the assumption that both the random effects and noise are normally distributed. It is important to note that in our application of CatBoost, we do not enforce any parametric modeling assumptions regarding the relationship between the features and the rewards.

Models	Utilize $\boldsymbol{x}$	Heterogeneity	Linear Assumption
MCMAB (ours)	✓	✓	
FD	✓		
FA-ind		✓	
Han2021	✓		✓

Table 1: MCMAB and baseline approaches.

To mimic the concurrent scenario, where multiple campaigns run simultaneously, we construct 50 distinct campaigns (" = 50), each with five randomly selected ad lines (= 5) and a daily budget limit of \$300 (< = 300). Budgets were distributed daily across ad lines within each campaign separately. The stochastic observations for the total clicks are generated using the base model (3) with $6, f_<, f_n$ as parameters. Similarly, to mimic the sequential scenario, where campaigns come in sequence, each campaign is randomly constructed with five ad lines (= 5) and lasts 50 days () = 50) with a daily budget of \$300 (< = 300).

Baselines. Our studies compare ten approaches, including the proposed MCMAB algorithm with LR, NN, and GP as working models to model the relationship between ;>61\<,:,0° and features. We also examine the feature-determined (FD) counterpart of each version of the MCMAB algorithm, which shares the MCMAB's modeling assumptions but with $f_<$ = 0. LR-based approaches use feature information $\boldsymbol{x}_{<,:}$, encompassing channels and supply source of the corresponding ad line, and the budget limit of the corresponding campaign. They assume that $61\boldsymbol{x}_{<,:},0° = q1\boldsymbol{x}_{<,:},0°)$ \$, with q being a deterministic function that transforms the tuple information $1\boldsymbol{x}_{<,:},0°$ to further include interaction terms between channels, supply sources, and the logarithm of budget shares. GP-based approaches assume $61\boldsymbol{x}_{<,:},0°$ follows a Gaussian Process, while NN-based approaches employ a fully connected 3-layer neural network 6 with a width of 30 for the concurrent setting and a width of 26 for the sequential setting. Additionally, we explored *Hibou*, the current method used in the ADSP system, which posits linear relationships between ad-line performance and assigned budgets, allocating budgets solely on estimated gradients without further exploration or information sharing. The study also includes an evaluation of *Han2021_RF* and *Han2021_LR*, the contextual bandits approaches from Han and Arndt [4], utilizing Random Forest and Linear Regression as the global model, respectively. The local model for each ad line is fitted as a Bayesian linear model using an augmented dataset consisting of 30 predicted returns generated by the fitted global model and the ad line's observed history. Lastly, we considered *FA-ind*, a baseline approach that independently learns the distribution of each \<,:,0. See Table 1 for a summary.

To ensure a fair comparison, we applied uninformative priors for all methods. Specifically, for *FD-LR* and *MCMAB-LR*, we used \$ N 10, 200°; for *FD-GP* and *MCMAB-GP*, we used zero-mean priors with RBF kernels; for *FD-NN* and *MCMAB-NN*, we initialized the networks with all weights sampled from normal distributions with zero mean Zhang et al. [19]; for *Hibou*, we started with an even allocation; and for *Han2021_LR* and *Han2021_RF*, we used N 10, 200° as the prior for local model parameters[4].

Results. Figure 3 depicts the average reward (i.e., the average number of clicks) received after implementing the budget allocation strategies suggested by each approach. Overall, feature-guided approaches (*MCMAB-LR*, *MCMAB-GP*, and *MCMAB-NN*) demonstrated greater average reward, outperforming other methods. Compared to *Hibou*, *MCMAB* showed an approximate 18% increase in the average number of clicks obtained at the conclusion of the experiment in the concurrent setting, and a 16% increase in the sequential setting.

Failing to utilize any feature information, *FA-ind* struggles with the curse of dimensionality and limited interaction opportunities, resulting in a significantly slower learning process with the lowest average reward, for both settings. Under the concurrent setting, *Hibou*, which uses only the budget information and learns the reward distribution for each ad line independently, continues to show a lower average reward than approaches that utilize additional ad line metadata information. On the other hand, feature-determined approaches (*FD-LR*, *FD-GP*) and *Han2021* initially outperform feature-guided approaches (*MCMAB-LR*, *MCMAB-GP*) but ultimately sustain lower average reward due to their restricted model assumptions. It should be noted that because the current network structure is naively specified without carefully fine-tuning its width and depth, *FD-NN* and *MCMAB-NN* perform worse than other feature-determined approaches, indicating that the current network structure fails to capture the relationship's complexity well. We could expect that the NN-based approach will perform better with further fine-tuning.

Under the sequential setting, *Hibou* demonstrates superior performance during the initial stages. This is because when campaigns are introduced sequentially, the metadata information is limited initially, impeding reasonable estimations for other feature-based approaches. As the system accumulates data from an increasingly diverse range of campaigns, the average reward for approaches that utilize metadata for information sharing shows a marked increase. In contrast, *Hibou*'s average reward converges to be constant, reflecting its inability to leverage learnings from past campaigns. Similar to what we observed in the concurrent setting, feature-determined approaches and *Han2021* yield a lower average reward compared to *MCMAB*. This underperformance is primarily attributed to their failure to adequately address the heterogeneity among base-arms.

Finally, *MCMAB-LR* performs better than *MCMAB-NN* and *MCMAB-GP* under the concurrent setting. This is presumably due to the linear properties of the dataset we used, which reduce the efficacy of the more complex GP and NN models, potentially leading to overfitting. In contrast, *MCMAB-GP* performs better than *MCMAB-LR* and *MCMAB-NN* under the sequential setting, with *MCMAB-LR* gradually approaching the performance level of *MCMAB-GP* as more campaigns are completed. This is mainly due to the initial scarcity of metadata, which hinders *MCMAB-LR*'s ability to establish a reliable linear model, whereas the Gaussian process, by using its kernel function, can effectively focus on more relevant features and ignore features containing less information. It is worth noting that fine-tuning kernels and hyperparameters can improve the performance of GP-based approaches, while the performance of NN-based approaches can be enhanced by further adjusting the neural network structure and learning rate.