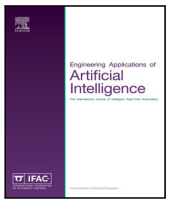
Engineering Applications of Artificial Intelligence 164 (2026) 113200

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai



Research paper

Hierarchical Bayesian Deep Learning for return on advertising spend prediction: A probabilistic approach to e-commerce advertising

Arti Jha [a], Ashutosh Bhatia [a], Kamlesh Tiwari [a], Hari Mohan Pandey [b,*]

[a] Department of Computer Science and Information Systems, BITS Pilani, Rajasthan, India
[b] Department of Computing Informatics, Bournemouth University, Poole, Dorset, BH12 5BB, UK

ARTICLE INFO

Keywords:
Return on Advertising Spend
Probabilistic forecasting
Deep learning
Uncertainty quantification
Online advertising optimization

ABSTRACT

In the highly competitive landscape of e-commerce advertising, maximizing return on advertising spend (ROAS) is crucial yet inherently uncertain due to auction-based bidding dynamics and fluctuating market conditions. Traditional deterministic models struggle to capture this uncertainty, necessitating a probabilistic approach that balances predictive accuracy with interpretability. To address this challenge, the paper proposes a novel Hierarchical Bayesian Deep Learning framework. The architecture was motivated by initial exploratory analysis using a Bayesian Belief Network (BBN) to map structural dependencies, while the final deep learning model overcomes scalability limitations using self-attention mechanisms and a Mixture Density Network (MDN) for full distributional modeling of ROAS. The BBN captures dependencies among campaign variables, enhancing interpretability, while the hierarchical deep learning architecture leverages self-attention mechanisms to address scalability challenges in high-dimensional settings. Experimental results reveal that the proposed framework achieves 22.8% lower RMSE and 27.4% better Negative Log Likelihood (NLL) and up to 31.2% lower Kullback–Leibler divergence (KLD) than state-of-the-art methods (DeepAR, Prophet, NGBoost), achieving an R^2 of 98% with an inference speed of 5.2 ms per campaign, confirming its feasibility for real-time bidding applications which typically require sub-10ms latency, enabling a feasible real-time bidding. Ablation studies confirm that attention-driven feature selection and calibrated uncertainty quantification significantly enhance both predictive performance and explainability, identifying key drivers of campaign success. By providing precise, uncertainty-aware, and explainable predictions, this approach enables adaptive bidding strategies, optimized budget allocation, and risk management, setting a new benchmark for intelligent decision-making in digital advertising.

1. Introduction

In the rapidly evolving domain of online advertising, businesses strive to utilize targeted strategies to maximize their Return on Investment (ROI) (Sutton and Barto, 2018). The unique challenges of e-commerce platforms, however, remain underexplored. Predicting ROAS in this domain is particularly difficult due to a confluence of factors: high-dimensional sparse interactions between thousands of campaign settings; a fundamentally stochastic auction environment where outcomes are probabilistic; and delayed, noisy conversion signals. These challenges, exemplified by Amazon's Sponsored Ads, create significant uncertainty and render traditional deterministic models inadequate (Qu et al., 2019; Zhou et al., 2022a). In this high-stakes financial environment, traditional deterministic models often underperform, as they provide single-point forecasts for metrics like Return on Advertising Spend (ROAS) and cannot account for the inherently probabilistic nature of auction outcomes, fluctuating market conditions, or competitor bidding (Kumari and Toshniwal, 2021; Park and Lee, 2022). A probabilistic approach is therefore essential for strategic and risk-aware decision-making (Panda et al., 2024).

Recent research has advanced core advertising objectives like bid optimization and click-through rate prediction using techniques such as reinforcement learning (Yakovleva et al., 2024; Chen et al., 2021) and transformers (Jiang et al., 2018; Mao et al., 2023). However, these approaches often culminate in deterministic predictions, exhibiting overconfidence and a lack of robust uncertainty quantification, which is critical for budgeting under risk (Gal and Ghahramani, 2016; Rahaman et al., 2021). Recognizing this gap, the latest research has

* Corresponding author.
E-mail addresses: p20210471@pilani.bits-pilani.ac.in (A. Jha), ashutosh.bhatia@pilani.bits-pilani.ac.in (A. Bhatia), kamlesh.tiwari@pilani.bits-pilani.ac.in (K. Tiwari), hpandey@bournemouth.ac.uk (H.M. Pandey).

https://doi.org/10.1016/j.engappai.2025.113200
Received 31 March 2025; Received in revised form 27 August 2025; Accepted 12 November 2025
Available online 18 November 2025

begun to pivot towards explicit uncertainty modeling. For instance, recent studies have applied reinforcement learning to cost-per-acquisition optimization with probabilistic bounds (Mao et al., 2023), and contextual bandits have been developed for more nuanced bidding strategies (Wang et al., 2022). Furthermore, the development of specialized probabilistic time-series models for revenue forecasting (Katsman et al., 2023) and new theoretical frameworks for distributional prediction (Li et al., 2024) highlight a clear trajectory in the field. Yet, a comprehensive and scalable solution for full distributional ROAS forecasting in e-commerce, which is crucial for practical budget allocation, remains an open challenge (Chen et al., 2023).

To overcome these limitations, this paper presents a hierarchical Bayesian deep learning model that integrates self-attention and a Mixture Density Network (MDN) output layer. Instead of predicting a single expected value, this approach models the entire distribution of ROAS, providing calibrated predictions and deeper insights into campaign variability. Stochastic Weight Averaging (SWA) is combined with Bayesian inference to deliver robust uncertainty estimates while maintaining computational efficiency, and self-attention captures long-range dependencies across temporal and campaign-specific features (Park and Lee, 2022). Unlike prior methods that rely on point estimates and fail to quantify risk, the proposed model leverages Bayesian inference and self-attention to generate full probability distributions over ROAS, enabling risk-aware and explainable decision-making in digital advertising. Explainability is prioritized through interpretability, including attention-based weight distribution and feature importance metrics, allowing advertisers to understand how key factors—such as impressions, cost per click, and targeting keywords—influence ROAS fluctuations. This transparency fosters trust in automated solutions, particularly when high-stakes budgetary decisions depend on model outputs. To rigorously validate the proposed model, the experimental setup accounts for multiple factors, including seasonal variations (e.g., Black Friday, Cyber Monday), distribution shifts, and changes in user engagement trends, ensuring a realistic evaluation of bidding strategies within Amazon's ecosystem. Based on the preceding discussion, the key contributions of this paper are as follows:

Key contributions

1. First, we propose a Bayesian Self-Attention architecture that models the complete ROAS distribution, offering calibrated *uncertainty estimates* to support informed budget decisions. The framework achieves competitive performance — demonstrating lower RMSE, higher R^2, and fast inference times — ensuring feasibility for real-time advertising applications.
2. Second, we employ a Bayesian Belief Network (BBN) to capture variable dependencies, leveraging Conditional Probability Distributions for structured probabilistic reasoning and enhanced interpretability.
3. Third, we develop an explainability framework that incorporates attention maps and weight distribution analysis, providing clear insights into how various features influence campaign outcomes.
4. Finally, we conduct practical validation on large-scale e-commerce data, demonstrating the model's feasibility in complex, high-dimensional, and dynamic advertising environments.

The remainder of this paper is structured as follows: Section 2 reviews the related literature, Section 3 details the proposed model architecture, Section 4 presents experimental evaluation results, Section 5 discusses model explainability and interpretability, and Section 6 concludes the paper with key findings and future research directions.

2. Background and related work

This section outlines how e-commerce platforms generate and process advertising data, followed by an exploration of why probabilistic deep learning offers an advantage over traditional methods.

2.1. Advertisement campaigns in ecommerce

Advertising in e-commerce is a dynamic and data-driven field, where success hinges on how effectively businesses process and act on consumer behavior data (Danaher et al., 2010). Every click, search, and purchase generates a signal that can be used to optimize ad targeting and bidding strategies.

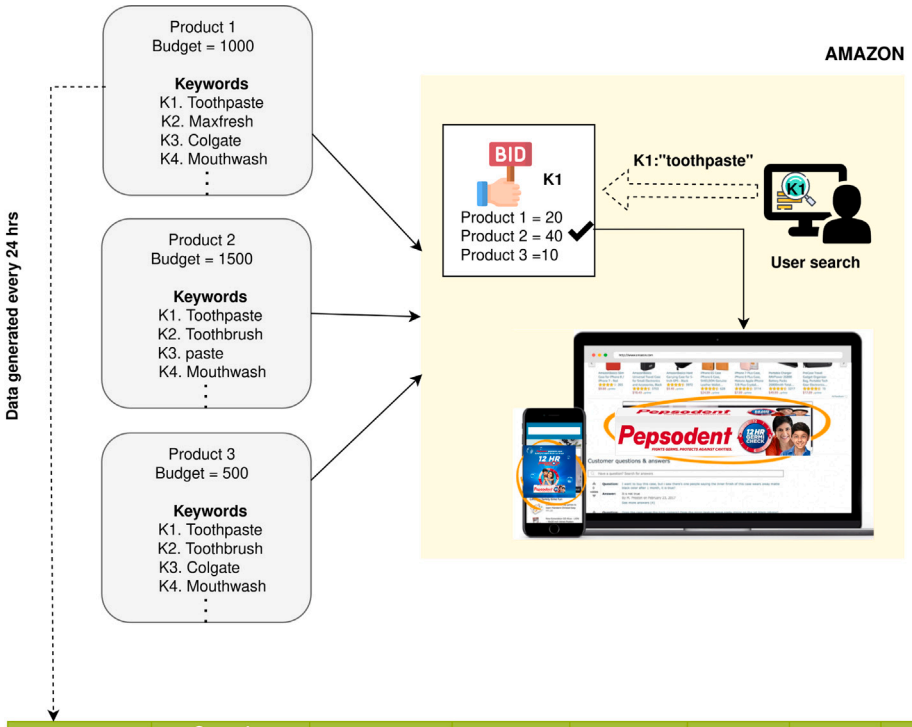
Amazon, as a leading online marketplace, operates a PPC (Pay-Per-Click) auction-based advertisement system, where sellers only pay a fee when a user actually clicks on their ad. In this system, sellers compete for premium search placements by bidding on relevant keywords. This system enables advertisers to strategically allocate budgets and optimize bids to ensure their products appear prominently in search results when customers search for relevant items. However, the dynamic nature of consumer behavior, seasonal market trends, and fluctuating competition introduce significant challenges in designing effective bidding strategies. Amazon offers multiple types of advertising campaigns, including *Sponsored Products Ads*, *Sponsored Brands Ads*, and *Sponsored Display Ads*. Among these, Sponsored Brands and Sponsored Products campaigns are particularly relevant for search-driven conversions, as they rely on competitive keyword auctions. Advertisers bid on specific keywords, and the highest bidder secures the most visible ad placement. However, the final cost incurred by the winning advertiser is determined through a second-price auction mechanism, where the highest bidder pays only the amount bid by the second-highest competitor. This ensures cost-efficient ad pricing while maintaining competitive placement. Fig. 1 illustrates the hierarchical structure of Amazon's bidding system, where multiple brands compete for top search rankings based on keyword-based auctions. In this system, advertisers are charged on a *Cost-Per-Click (CPC)* basis, where costs are deducted from pre-allocated budgets upon user interaction with the ad.

2.2. Data generation & processing pipeline

Modern e-commerce platforms generate vast amounts of behavioral data, tracking everything from product views and cart additions to ad impressions and click-through rates (CTR). This data serves as the foundation for targeted advertising, where machine learning models attempt to predict which ads will be most effective for each user (Cavalcante et al., 2016). However, raw behavioral logs alone are not enough—real-time data processing is essential to extract meaningful patterns. Advertising data typically flows through a pipeline that begins with collection and preprocessing. User activity is captured through cookies, tracking pixels, and API integrations and then refined to remove noise and inconsistencies. Relevant features are engineered to provide contextual insights, such as the time of day affecting shopping habits or how browsing sequences indicate purchase intent. Advanced ad platforms use streaming architectures to process this data in real-time, ensuring that models are constantly updated with the latest consumer signals. Once the data is structured, machine learning models use it to drive ad optimization. They predict click probabilities, segment audiences, and dynamically adjust bidding strategies in real-time advertising markets. A key challenge in this process is uncertainty, consumer behavior is not static, and a model that relies too heavily on past data may fail when trends shift. This is where probabilistic deep learning becomes valuable.

2.3. Probabilistic deep learning in ad optimization

Deterministic deep learning models are ill-suited for the dynamic nature of e-commerce advertising, as their overconfident point-estimates can lead to poor budget allocation. Bayesian deep learning addresses this by treating model parameters as distributions to produce probabilistic forecasts that quantify uncertainty. This is essential for risk-aware, adaptive bidding. Furthermore, Bayesian models can



Campaign ID	Campaign Name	Keyword ID	Keyword Text	Impressions	Clicks	Cost	Sales
45306679061110	Oral Care - Hero - Hum Electric TB	70220220564144	toothbrush	444632	2733	72222.09	33322.53
45306679061110	Oral Care - Hero - Hum Electric TB	80702627762597	toothbrush	380425	425	3286.71	3186.53
89658563464577	Oral Care - Category - Toothpaste	64229905140852	toothpaste	264320	105	1006.07	287.6
89658563464577	Oral Care - Category - Toothpaste	64229905140852	toothpaste	262259	112	1110.61	337.05

Fig. 1. Hierarchy of Amazon's Sponsored Advertisement bidding process.

be efficiently updated with new data via posterior updating, making them ideal for changing market conditions (Wilson and Izmailov, 2020b; Lasowski and Nolde, 2021; Deshpande et al., 2022; Masegosa et al., 2021). Bayesian models also offer superior generalization in data-poor situations, which reduces overfitting. Our work combines this with a self-attention mechanism to facilitate robust probabilistic reasoning for real-time applications, such as dynamic audience segmentation and cost-efficiency optimization (Deshpande et al., 2022; Masegosa et al., 2021; Wilson and Izmailov, 2020a). While proven in domains like healthcare (Ker et al., 2017) and finance, Bayesian deep learning is underexplored in advertising (Ghahramani, 2015; Wang and Yeung, 2020). This research fills that gap by presenting a unified framework that leverages both techniques to effectively manage risk and adapt to dynamic e-commerce environments (Polson and Sokolov, 2017; Patel et al., 2015).

2.4. Related work

Accurate Return on Ad Spend (ROAS) forecasting is crucial for optimizing budget allocation in e-commerce advertising. However, most existing methods rely on deterministic models, which fail to account for the uncertainty in auction-based bidding, user behavior, and market competition. This section reviews key research areas related to ad revenue prediction, highlighting their limitations and how the proposed model contributes to the field. Table 1 summarizes key contributions in online advertising optimization, detailing each work's focus, approach, datasets, metrics, findings, and limitations.

2.4.1. Revenue maximization and ad budget allocation

Prior work on revenue maximization has evolved from static models to sophisticated auction-based methods with adaptive bidding and multi-objective constraints (Yakovleva et al., 2024; Li et al., 2021; Qu et al., 2019; Akande and Haq, 2021; Chen et al., 2021; Mao et al., 2023; Wu and Chen, 2021). However, these approaches remain fundamentally deterministic, providing single-point ROAS predictions that ignore the uncertainty of dynamic ad campaigns. Our work addresses this limitation by introducing a probabilistic forecasting approach that enables risk-aware budget allocation.

2.4.2. Bid optimization and real-time bidding strategies

Bid optimization is crucial in balancing ad performance with cost efficiency (Yakovleva et al., 2019), yet most existing strategies remain deterministic. Bannour et al. (2023) developed data-driven bid adjustment mechanisms, and Zhang et al. (2021) explored predictive budget allocation models. While Jiang et al. (2018) applied Transformer-based architectures to model bid landscapes, and Wang et al. (2022) used deep Q-networks for real-time bidding, their models do not quantify prediction confidence. Since auction dynamics introduce high variance, deterministic models cannot capture risk-adjusted bidding decisions. Reinforcement learning (RL), which is primarily used in search engine real-time bidding. However, RL's immediate feedback loops limit its usefulness in e-commerce platforms with aggregated and delayed bid data (Zhou et al., 2022b; Jin et al., 2018). RL also struggles to capture non-linear interactions between metrics like CPC and ROAS due to post-auction data (Liu et al., 2024), and thus, a probabilistic approach is better suited to this scenario.

Table 1
Methodological comparison of related works in online advertising optimization.

Paper	Core Issue	Paradigm	Metrics	Contribution	Limitation
Kini and Manjunatha (2020)	Revenue maximization	Supervised (Multitask NN)	Revenue uplift, accuracy	Improved recommendations	No uncertainty; ignores market factors
Li et al. (2021)	Pricing strategies in IaaS	Market-Oriented Optimization	Profitability, efficiency	Flexible pricing models for revenue	Deterministic; lacks dynamic adjustment
Jiang et al. (2018)	Auction mechanism design	Game Theoretic (Data-Driven)	Bid efficiency, cost savings	Improved cloud resource allocation	Limited generalizability
Bannour et al. (2023)	Budget optimization	Predictive Regression	ROI, budget utilization	Enhanced budget allocation for audio ads	No uncertainty, not-validated on display ads
Yakovleva et al. (2024)	Bid landscape forecasting	Sequential (Transformer)	Accuracy, MSE	More accurate bid forecasting	Deterministic; computationally expensive
Gupta et al. (2022)	CTR prediction in RTB	Sequential (Dynamic NN)	AUC, log-loss	Improved real-time CTR prediction	Point-estimates only; struggles with sparsity
Agarwal et al. (2009)	Temporal CTR prediction	Spatio-Temporal Model	CTR, precision	Better handles time-dependent behavior	Deterministic; high computational overhead
Huang et al. (2019)	Feature engineering for CTR	Deep (Bilinear Feature Interaction)	AUC, accuracy	Improved feature interaction modeling	No uncertainty; requires large labeled data
Zhang et al. (2021)	Portfolio-based bidding	Optimization (Stochastic Control)	Cost efficiency, revenue gain	Enhanced bidding effectiveness	Point-estimates; not for price fluctuations
Balseiro et al. (2014)	Yield optimization	Economic (Mechanism Design)	Revenue max., fill rate	Better ad inventory allocation	Deterministic; ignores user engagement

2.4.3. Click-through rate prediction and ROAS estimation

CTR prediction plays a crucial role in estimating ad engagement (Jha et al., 2023), but traditional models lack uncertainty quantification. Gharaibeh et al. (2017) and Cai et al. (2018) developed dynamic neural networks for CTR forecasting, while Kumari and Toshniwal (2021) and Huang et al. (2019) explored spatio-temporal and feature-based interactions. Although these models improve CTR accuracy, they fail to express how uncertain their predictions are, which is critical when forecasting downstream revenue (ROAS). After the introduction of the attention mechanism by Zhang et al. (2014) and Chandra and He (2021) introduced self-attention with knowledge distillation to handle sparse datasets, but their approach still produces point estimates rather than probability distributions.

2.4.4. Auction mechanisms and adaptive bidding

Auction-based ad placements operate in highly volatile environments where bid prices and user engagement shift dynamically. Sutton and Barto (2018) and Rafieian and Yoganarasimhan (2021) studied ad exchange optimization, while Aronowich et al. (2014) developed auction-based revenue strategies. Although these studies have improved accuracy and efficiency, many lack real-time responsiveness to sudden market changes. Tiwari et al. (2023) addressed this concern by utilizing a deep Q-network for real-time bidding. However, these works do not explicitly model bid price uncertainty, limiting their ability to adjust bidding dynamically. Gal and Ghahramani (2016) proposed contextual bandit models for bid learning, but bandit approaches lack full probabilistic ROAS modeling, leading to uncertainty in ad budget allocation.

2.4.5. Bayesian deep learning in advertising

Despite recent advances in deep learning for advertising, uncertainty estimation remains largely unexplored. Most neural networks overfit past trends, failing to provide well-calibrated uncertainty measures. Akande and Haq (Šoltés et al., 2020) reviewed machine learning methods for ad optimization but highlighted the issue of deterministic overconfidence in model predictions. Amazon (2022) studied AI-driven CTR forecasting but did not incorporate uncertainty into revenue forecasting. Rahaman et al. (2021) used reinforcement learning for ad spend optimization, yet their models still produce deterministic bid recommendations. Bayesian deep learning overcomes these limitations by modeling distributions over ROAS outcomes, ensuring risk-aware and interpretable decision-making in ad campaigns.

2.4.6. Limitations of existing methods and contribution

While existing methods have improved ad optimization, their reliance on point-estimate predictions limits their effectiveness in highly volatile e-commerce environments. The proposed Bayesian Self-Attention model explicitly accounts for uncertainty by integrating Bayesian inference with attention mechanisms, leading to: (1) Risk-aware bidding and budget allocation: Unlike traditional deep learning models, our approach estimates the entire probability distribution over ROAS, providing confidence intervals instead of fixed-point forecasts. (2) Scalability for large-scale ad campaigns: Existing Bayesian models struggle with large-scale applications due to computational complexity, whereas our hierarchical Bayesian framework balances accuracy and efficiency. (3) Improved interpretability: Our model provides weight-distribution analysis and feature attribution mechanisms, ensuring transparency in ad spend decisions. By transitioning from deterministic forecasting to probabilistic modeling, this approach ensures more reliable and uncertainty-aware decision-making for advertisers.

3. Problem definition and proposed solution

In e-commerce, optimizing marketing strategies hinges on accurately forecasting ROAS, a vital metric for evaluating the effectiveness of advertising investments. A precise and reliable ROAS prediction model enables businesses to optimize budget allocation, refine bidding strategies, and maximize return on investment. Traditional deterministic models typically provide point estimates of ROAS, failing to account for the uncertainty surrounding revenue generation. The true distribution of ROAS is influenced by factors such as seasonality, market competition, evolving user behavior, and macroeconomic conditions. These factors introduce considerable variability, making it difficult for these models to adequately represent such complexity. Therefore, a more efficient and uncertainty-aware predictive model is required.

To address these challenges, a Deep Bayesian Neural Network (BNN) has been proposed for probabilistic ROAS prediction. This model integrates deep learning with Bayesian inference to estimate the full probability distribution of ROAS. Unlike conventional models that provide only point estimates, this approach outputs a comprehensive probability distribution, allowing decision-makers to assess predictive uncertainty and make more informed budgetary decisions. Such a model accounts for risks and opportunities, enhancing overall decision-making. Let the input feature vector at time t be denoted as $\mathbf{x}_t \in \mathbb{R}^d$, where d represents the number of predictive features. The goal is

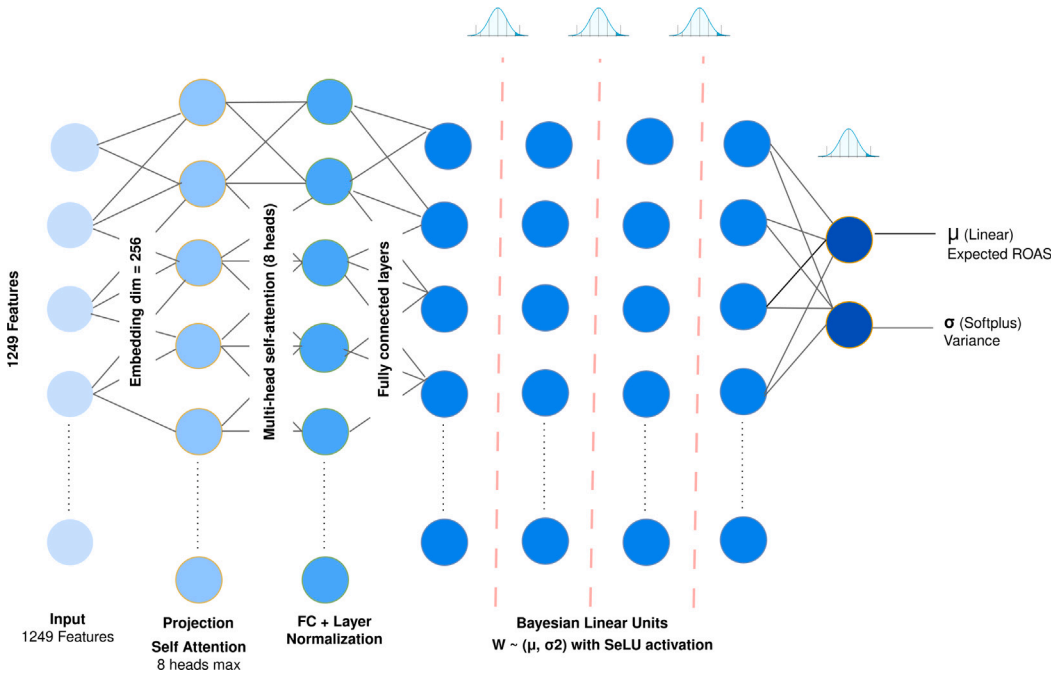


Fig. 2. Proposed Deep Bayesian Neural Network with self attention model architecture.

to estimate the conditional probability distribution of ROAS at time t, expressed as $p(y_t|\mathbf{x}_t)$, where y_t is the actual ROAS at time t. The approach involves using a Bayesian Neural Network to model this conditional distribution. Instead of learning fixed weights in the network, distributions over the weights are learned. This facilitates the quantification of epistemic uncertainty, which reflects uncertainty in the model's parameters due to limited data. The posterior distribution over the weights $\mathbf{W}$, given the input features $\mathbf{X}$ and observed ROAS values $\mathbf{Y}$, is represented as:

$$p(\mathbf{W}|\mathbf{X},\mathbf{Y})$$

To make the posterior computation feasible, the weights' distribution is approximated as a multivariate Gaussian:

$$p(\mathbf{W}|\mathbf{X},\mathbf{Y}) \approx \mathcal{N}(\boldsymbol{\mu}_{\text{BNN}}, \boldsymbol{\Sigma}_{\text{BNN}})$$

where $\boldsymbol{\mu}_{\text{BNN}}$ and $\boldsymbol{\Sigma}_{\text{BNN}}$ are the mean and covariance of the weights, respectively, estimated using stochastic optimization methods. The model is trained by minimizing a loss function that is formally derived from the Maximum A Posteriori (MAP) estimation framework. The MAP objective is to find the mode of the posterior distribution of the weights W given the data $\{X, Y\}$, which is equivalent to minimizing the negative log-posterior:

$$W_{\text{MAP}} = \arg\min_{W}[-\log p(Y|X,W) - \log p(W)] \tag{1}$$

Here, $-\log p(Y|X,W)$ is the negative log-likelihood, and $-\log p(W)$ is the negative log-prior. By choosing a zero-mean Gaussian prior for the weights, $p(W) \sim \mathcal{N}(0,(1/\lambda)I)$, the negative log-prior becomes proportional to $\lambda\|W\|^2$. This yields the final objective function used for optimization:

$$\mathcal{L}(W,X,Y) = -\log p(Y|X,W) + \lambda\|W\|^2 \tag{2}$$

The first term in Eq. (2) is the likelihood term, which drives predictive accuracy, while the second term is the L2 regularization (or weight decay) that stems from the Gaussian prior, preventing overfitting and encouraging simpler models. Our goal is to find not just this single point estimate W_{MAP}, but to approximate the full posterior distribution around this mode using Stochastic Weight Averaging (SWA), as detailed in Section 3.1. SWA is used to efficiently propagate uncertainty through the network. This technique averages the weights over multiple stochastic gradient descent (SGD) iterations, approximating the posterior distribution over the network's parameters. The combination of SWA and Bayesian inference allows for efficient uncertainty propagation, enabling faster predictions during inference while maintaining accurate uncertainty estimates. This model, trained iteratively, approximates the true ROAS distribution, ensuring that uncertainty in the prediction reflects the real-world variability observed in advertising scenarios. The Bayesian Neural Network captures not just the expected ROAS but also the entire distribution, aiding more robust decision-making in uncertain environments. Additionally, a self-attention mechanism is included in the architecture to capture long-range dependencies and sequential patterns, which are crucial for modeling time-dependent factors such as evolving user behavior or market dynamics. Unlike recurrent architectures, self-attention computes attention scores across all time steps in parallel, providing a global context for each time step. Mathematically, the self-attention mechanism operates as follows: given an input sequence of vectors $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, where each $\mathbf{x}_i \in \mathbb{R}^d$ represents an input at time step i, the attention mechanism computes the following:

For each time step i, the input sequence is transformed into query, key, and value vectors, defined as:

$$\mathbf{q}_i = \mathbf{W}_Q\mathbf{x}_i, \quad \mathbf{k}_i = \mathbf{W}_K\mathbf{x}_i, \quad \mathbf{v}_i = \mathbf{W}_V\mathbf{x}_i,$$

where $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V \in \mathbb{R}^{d\times d}$ are learned weight matrices for the query, key, and value projections.

The attention score between two time steps i and j is computed as:

$$\text{Attention Score}_{ij} = \frac{\mathbf{q}_i^T\mathbf{k}_j}{\sqrt{d}},$$

This score represents how much attention is given to $\mathbf{x}_j$ when processing $\mathbf{x}_i$.

The attention scores are then normalized using the softmax function:

$$\alpha_{ij} = \frac{\exp(\text{Attention Score}_{ij})}{\sum_{k=1}^{n}\exp(\text{Attention Score}_{ik})},$$

and

$$\alpha_{ij} = \text{softmax}(\text{Attention Score}_{ij}),$$

The output for $\mathbf{x}_i$ is computed as the weighted sum of the value vectors $\mathbf{v}_j$, with attention weights α_{ij} :

$$\mathbf{z}_i = \sum_{j=1}^{n} \alpha_{ij} \mathbf{v}_j.$$

This output $\mathbf{z}_i$ is then passed through a feedforward network and subjected to normalization and residual connections. A close representation of the neural network architecture for the proposed model has been illustrated in Fig. 2. The self-attention mechanism captures dependencies between time steps, providing a global context for each. The mechanism is repeated across multiple layers of the model, with each layer incorporating multi-head attention for more complex relationships. The attention mechanism is computationally efficient, as it enables parallelization and allows the model to learn from all parts of the input sequence.

3.1. Approximating the Bayesian posterior with Stochastic Weight Averaging

A central challenge in Bayesian deep learning is the intractable nature of the true posterior distribution over network weights, $p(W|X,Y)$. While methods such as Markov Chain Monte Carlo (MCMC) provide theoretical convergence guarantees, they remain computationally infeasible for deep models at scale. We employ Stochastic Weight Averaging (SWA) as a scalable and theoretically motivated method to approximate the Bayesian posterior.

Connection to Langevin dynamics and stationary distributions

The theoretical basis for SWA rests on the connection between stochastic gradient descent (SGD) and Langevin dynamics. As shown by Welling and Teh (2011), adding Gaussian noise to SGD yields Stochastic Gradient Langevin Dynamics (SGLD), which simulates samples from the posterior:

$$W_{t+1} = W_t + \frac{\eta_t}{2} \nabla \log p(W_t|X,Y) + \varepsilon_t, \quad \varepsilon_t \sim \mathcal{N}(0, \eta_t), \tag{3}$$

Even without explicit noise, SGD with mini-batches exhibits gradient noise that under standard assumptions (e.g., constant learning rate, smooth loss landscape) can be modeled as a discretized Ornstein–Uhlenbeck (OU) process (Mandt et al., 2017):

$$dW_t = -A(W_t - \mu)dt + \sqrt{2D}\, dB_t, \tag{4}$$

The stationary distribution of this process is Gaussian:

$$p(W) \propto \exp\left(-\frac{1}{2}(W-\mu)^\top \Sigma^{-1}(W-\mu)\right), \tag{5}$$

where μ is the mean around a mode of the loss, and Σ reflects the noise-induced covariance.

SWA as a consistent estimator of posterior mean

SWA averages weights across SGD trajectories:

$$\mu_{\text{BNN}} = W_{\text{SWA}} = \frac{1}{T} \sum_{i=1}^{T} W_i, \tag{6}$$

Under the assumption that the SGD iterates $\{W_i\}$ are drawn from an ergodic Markov chain sampling the stationary distribution of the OU process, the SWA mean converges almost surely to the posterior mean:

$$\lim_{T\to\infty} \mu_{\text{BNN}} = \mathbb{E}_{p(W|X,Y)}[W]. \tag{7}$$

This is a direct consequence of the law of large numbers for ergodic processes.

Posterior approximation with diagonal Gaussian

We construct a Gaussian approximation to the posterior with mean μ_{BNN} and diagonal covariance $\text{diag}(\sigma^2_{\text{BNN}})$:

$$q(W) \approx \mathcal{N}(W \mid \mu_{\text{BNN}}, \text{diag}(\sigma^2_{\text{BNN}})), \tag{8}$$

The variance σ^2_{BNN} is estimated either from the empirical second moment of the collected weights or treated as a tunable hyperparameter. This posterior matches the form of the Gaussian stationary distribution induced by Langevin dynamics and has been empirically validated in related work.

Convergence and practical implications

We implement SWA by collecting model weights over the final 25% of training epochs using a high, constant learning rate to encourage exploration of a flat posterior mode. While this approximation does not match the exact posterior in full generality, it captures key structural properties, such as mode centering, local uncertainty, and flatness—that are sufficient for well-calibrated uncertainty estimates and robust downstream decision-making. This method offers a tractable alternative to MCMC or full variational methods, with competitive empirical performance and theoretical grounding.

4. Proposed model

The proposed framework is a hierarchical deep learning model designed for end-to-end probabilistic forecasting of ROAS. The term 'hierarchical' refers to the model's architectural structure, which processes information in two distinct stages:

1. A Representation Learning Layer: At the base of the hierarchy, a multi-head self-attention mechanism acts as a powerful feature encoder. Its unique contribution is the ability to model complex, non-local dependencies across the entire feature set. For instance, it can learn how a change in "targeting_keyword_A" dynamically influences the effectiveness of "budget_for_campaign_B", a task where traditional models struggle. It produces a dense, context-aware vector that captures these rich interactions.
2. A Probabilistic Regression Layer: At the top of the hierarchy, a Bayesian neural network takes the feature vector from the attention layer and performs probabilistic regression. This layer culminates in a Mixture Density Network (MDN) head, which outputs the full probability distribution of ROAS.

This two-stage hierarchy allows the model to first learn *what* features are important in context, and then to quantify the uncertainty associated with predicting an outcome based on those features.

4.1. Dynamic feature extraction and representation

Accurately predicting ROAS requires transforming raw advertising data into structured numerical representations that effectively capture underlying patterns. To achieve this, a multi-stage feature extraction pipeline is developed to ensure numerical stability, model temporal dependencies, and enhance the generalization capability of the network. The first stage involves numerical stabilization, where key features such as Click-Through Rate (CTR), Cost-Per-Click (CPC), and ad spending are normalized to maintain consistency and prevent scale-dependent biases. Following this, a temporal pattern modeling mechanism is applied, incorporating moving averages, lag-based transformations, and frequency-domain decompositions to capture seasonal variations and campaign trends. A multi-campaign embedding strategy is then applied, encoding campaigns into a latent space that enables cross-campaign knowledge transfer while preserving campaign-specific distinctions. Finally, a feature refinement process systematically eliminates redundant attributes using an information-theoretic selection criterion, ensuring

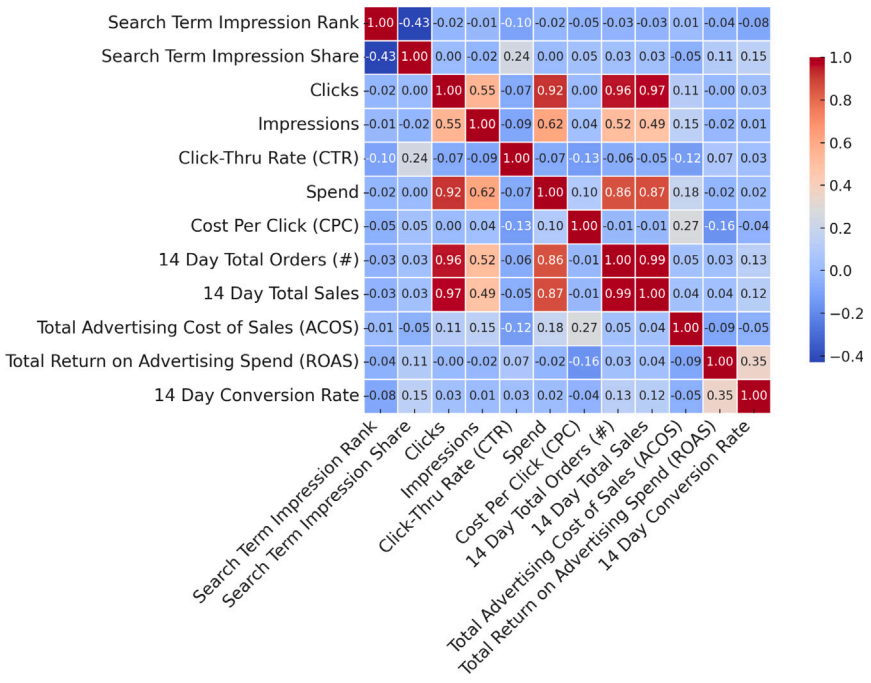


Fig. 3. Correlation matrix of key features.

that only the most predictive features contribute to the final model. This structured feature engineering process ensures that the input representations provide a stable and informative basis for subsequent probabilistic modeling.

4.2. Hierarchical probabilistic representation

At the core of the proposed architecture is a hierarchical Bayesian learning model that extends traditional deep neural networks by incorporating uncertainty-aware probabilistic modeling. The proposed model represents network weights as probability distributions, allowing it to quantify epistemic uncertainty in decision-making. The weight parameters are formulated as:

$$W \sim \mathcal{N}(\mu, \sigma^2), \tag{9}$$

where W denotes the set of learnable weights, μ represents the mean, and σ^2 captures the variance, thereby encoding the model's uncertainty. To efficiently capture complex cross-campaign interactions, the proposed model integrates an attention-driven dependency modeling mechanism. This mechanism assigns dynamic attention scores to past and concurrent campaigns, ensuring that the model focuses on the most relevant historical events when generating predictions. The attention mechanism is defined as:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \tag{10}$$

where Q, K, V represents the query, key, and value matrices, and d_k corresponds to the dimensionality of the key vectors. This hierarchical probabilistic model, coupled with attention-based contextual awareness, enables the model to learn both global and local dependencies within multi-campaign data (see Fig. 3).

The interaction between the self-attention mechanism and the subsequent Bayesian layers is fundamental to the model's hierarchical approach. The self-attention layers act as a powerful, data-driven feature extractor. They learn to dynamically re-weight and combine input features based on their contextual relevance, producing a rich, latent representation of the campaign's state. This representation, which captures complex temporal and cross-feature dependencies, is then fed into the fully-connected layers where the weights are treated as Bayesian random variables. In this framework, the Bayesian layers are not learning from the raw, sparse inputs, but from the dense, context-aware embeddings generated by the attention block. This separation of concerns — using attention for deterministic feature representation and Bayesian layers for probabilistic regression on those features — allows the model to handle high-dimensional input effectively while still providing robust uncertainty quantification over the final prediction.

4.3. Uncertainty-aware predictive modeling

The predictive layer of the proposed architecture is designed to generate full probability distributions over ROAS rather than point estimates. The choice of the distributional form is a critical modeling decision that must be justified by the empirical properties of the data. To provide this justification, we analyzed the statistical distribution of the target variable from our dataset. Fig. 4 shows the Kernel Density Estimate (KDE) of the log-transformed ROAS values. The plot provides compelling evidence that a simple unimodal distribution would be an inadequate choice. The distribution is distinctly multimodal, featuring a primary mode near a log-ROAS of 1.5, a significant secondary mode around 3.5, and a third, wider mode corresponding to very high-performing campaigns near 6.0. This structure strongly suggests the existence of several different underlying campaign archetypes or generative processes. For example, the main peak may correspond to standard, business-as-usual campaigns, while the other peaks could represent more successful, niche targeting strategies or campaigns benefiting from seasonal trends. A single Gaussian, skewed, or heavy-tailed distribution would fail to capture these distinct sub-populations. A Mixture Density Network (MDN), however, is a universal approximator of densities and is ideally suited to model such complex, multimodal data by assigning different Gaussian components to capture each mode. This provides a flexible and data-driven approach to modeling ROAS. Therefore, we model the conditional distribution of ROAS as a weighted sum of Gaussian components:

To reinforce this visual evidence, we performed a statistical comparison between Gaussian Mixture Models (GMMs) and alternative distributional assumptions, including skew-normal and Student's t-mixture models. The Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC) were used to evaluate model fit. As summarized

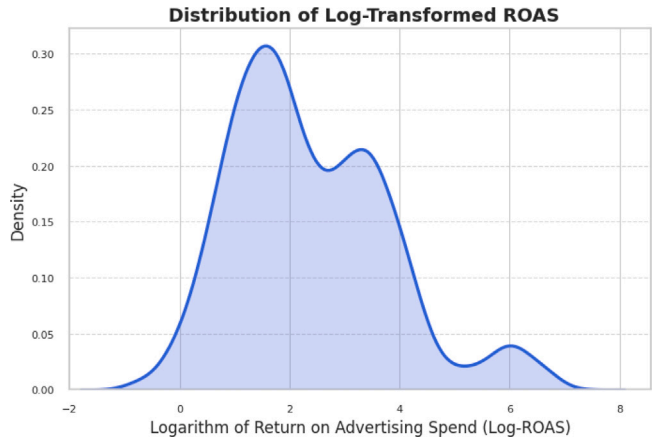


Fig. 4. Kernel Density Estimate (KDE) of the log-transformed Return on Advertising Spend (ROAS) from the experimental dataset. The presence of multiple modes (peaks) clearly indicates that a single probability distribution (e.g., a single Gaussian) would be insufficient to model the underlying data structure. This multimodality provides strong empirical justification for our choice of a Mixture Density Network (MDN).

Table 2
Model comparison for log-ROAS distribution fit.

Model	AIC ↓	BIC ↓	Log-Likelihood ↑	Wasserstein distance ↓
Gaussian mixture (3 comp.)	**1273.6**	**1312.8**	**−621.8**	**0.024**
Skew-normal distribution	1365.2	1379.1	−673.6	0.062
Student's t mixture (2 comp.)	1294.4	1338.7	−635.2	0.045

in Table 2, the GMM achieved the lowest AIC and BIC scores, indicating a better trade-off between model complexity and data fidelity. Additionally, the GMM exhibited the highest log-likelihood and the lowest Wasserstein distance to the empirical distribution. These results quantitatively confirm that mixtures of Gaussians are well-suited to model the observed log-ROAS distribution, offering superior fit over both skewed and heavy-tailed alternatives.

The predictive layer of the proposed architecture is designed to generate full probability distributions over ROAS rather than point estimates. To achieve this, a Mixture Density Network (MDN) head is used. Instead of outputting a single value, the final layer of our neural network is designed to output the parameters of a Gaussian Mixture Model conditioned on the input features x. Specifically, the network learns to predict the mixture weights $\pi_i(x)$, means $\mu_i(x)$, and variances $\sigma_i^2(x)$ for a predefined number of K Gaussian components:

$$p(y|x) = \sum_{i=1}^{K} \pi_i(x)\mathcal{N}(y|\mu_i(x), \sigma_i^2(x)), \tag{11}$$

where $\pi_i(x)$ denotes the mixture coefficient for the ith Gaussian component, $\mu_i(x)$ represents the component mean, and $\sigma_i^2(x)$ defines the variance, quantifying the uncertainty in prediction. The model is optimized by maximizing the log-likelihood of observed ROAS values under the estimated probability distribution:

$$\mathcal{L} = \sum_{t=1}^{T} \log\left(\sum_{i=1}^{K} \pi_i(x_t)\mathcal{N}(y_t|\mu_i(x_t), \sigma_i^2(x_t))\right), \tag{12}$$

where T denotes the number of training instances. These parameters are not manually initialized; they are the direct outputs of the final neural network layer and are optimized implicitly by minimizing the model's primary objective function: the Negative Log-Likelihood of the data given the predicted distribution. The incorporation of a probabilistic output layer enhances the model's capability to generate predictions under varying levels of market uncertainty. The estimated uncertainty values provide additional insight into prediction confidence, allowing advertisers to make risk-calibrated budget allocation decisions. The explainability of the proposed model is reinforced through structured visualization and interpretability techniques. The weight distribution analysis (Fig. 8a) ensures that learning is well-regulated across layers, avoiding excessive reliance on individual parameters. The attention-based dependency modeling (Fig. 7a) highlights the extent to which past campaigns influence current predictions, offering greater transparency in decision-making (see Table 3).

Table 3
Statistical summary of key numerical features.

Feature	Mean	Std Dev	Min	Max
Search term impression rank	3.06	6.37	1.0	266.0
Clicks	3.12	16.37	1.0	1181.0
Impressions	462.45	3326.49	0.0	398 280.0
Total orders	1.37	8.99	0.0	623.0
ROAS	20.23	112.43	0.0	8104.0

4.4. Training and inference procedure

To ensure clarity and reproducibility, we formalize the training and inference processes of our proposed Hierarchical Bayesian Deep Learning model in Algorithms 1 and 2, respectively. The inference procedure, detailed in Algorithm 2, utilizes the trained SWA model to generate a full probabilistic forecast for a new, unseen campaign instance.

Algorithm 1 Model Training Procedure

Require: Training dataset $D = \{X_{\text{train}}, Y_{\text{train}}\}$; Epochs E; Batch size B.
Require: Optimizer $\mathcal{O}$; Learning rate η; SWA start epoch E_{SWA}; SWA learning rate η_{SWA}.
Initialize model parameters W for model M.
Initialize SWA model M_{SWA}.
for epoch $e = 1$ to E **do**
  Partition D into mini-batches $\{x_b, y_b\}_{b=1}^{N}$.
  for each mini-batch $\{x_b, y_b\}$ **do** ▷ — Forward Pass —
    Predict mixture parameters: $\{\hat{\pi}_b, \hat{\mu}_b, \hat{\sigma}_b\} \leftarrow M(x_b; W)$. ▷ — Loss Calculation & Gradient Update —
    Compute NLL loss $\mathcal{L}_{\text{NLL}}(y_b, \{\hat{\pi}_b, \hat{\mu}_b, \hat{\sigma}_b\})$.
    Update weights $W \leftarrow \mathcal{O}(W, \nabla_W \mathcal{L}_{\text{NLL}})$.
  end for
  if $e \geq E_{\text{SWA}}$ **then**
    Update SWA weights W_{SWA} with current weights W. ▷ Collect weights for averaging
  end if
  Update learning rate schedule for η.
end for
Ensure: Trained SWA model M_{SWA} with parameters W_{SWA}.

5. Experimental results

The experimental evaluation is conducted on a large-scale e-commerce advertising dataset to rigorously validate the performance of our proposed model against established baselines.

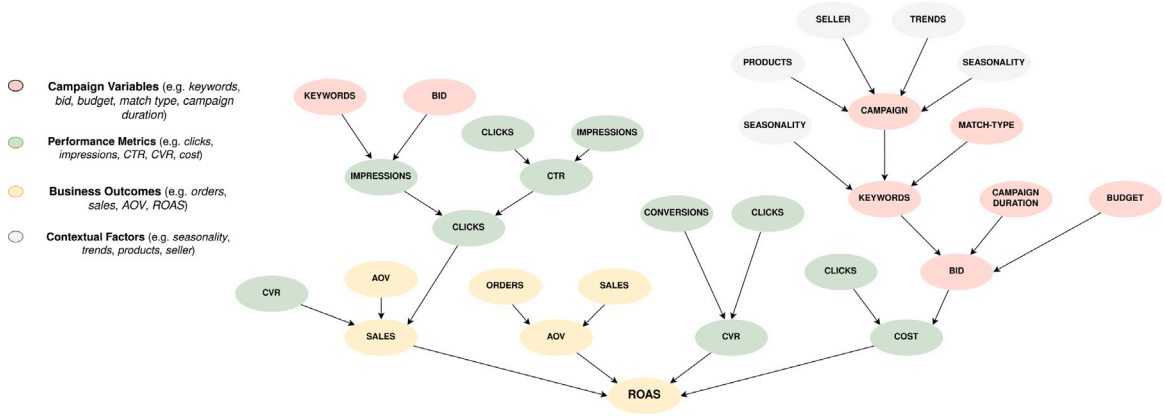


Fig. 5. Proposed Bayesian Belief Network (BBN) structure for ROAS forecasting.

Algorithm 2 Inference Procedure

Require: Trained SWA model M_{SWA}; a new campaign feature vector x_{new}.

// — Probabilistic Prediction —
Predict mixture parameters $\{\pi, \mu, \sigma\} \leftarrow M_{\text{SWA}}(x_{\text{new}})$.
// — Output Generation —
The parameters $\{\pi, \mu, \sigma\}$ define the conditional probability distribution $p(y|x_{\text{new}})$.
▷ From this distribution, various quantities can be derived:
- **Point Estimate:** Compute expected value $\mathbb{E}[y|x_{\text{new}}]$.
- **Full Posterior:** Draw samples $y_i \sim p(y|x_{\text{new}})$.
- **Risk Assessment:** Compute confidence or credible intervals.
return Predictive distribution parameters $\{\pi, \mu, \sigma\}$.

5.1. Dataset and preprocessing

The experimental evaluation is conducted on a large-scale e-commerce advertising dataset comprising 160,621 campaign instances spanning a 24-month timeframe. These campaigns encompass a wide array of objectives, budget allocations, and user segments, offering a comprehensive representation of digital advertising complexities.

5.1.1. Data source and collection

The proprietary dataset used in this study was collected from the Amazon Advertising API (v3.2) and comprises daily performance logs for 160,621 Sponsored Product and Sponsored Brand campaigns. The data spans a 24-month period from July 1, 2023, to June 30, 2025, a timeframe that includes multiple major shopping events (e.g., Black Friday, Cyber Monday) and diverse market conditions. Initial ingestion and aggregation were performed in a Snowflake data warehouse hosted on AWS. For modeling, we filtered out low-activity campaigns (fewer than 50 clicks over their lifetime) and removed records with clear data anomalies (e.g., non-zero clicks with zero impressions) to ensure data quality.

5.1.2. Feature engineering

The raw data includes 20 features related to campaign metadata and performance. We engineered the following key attributes and performed preprocessing steps to create the final feature set:

- **Derived Metrics:** Core performance indicators were computed, including Click-Through Rate (CTR) = Clicks/Impressions, and the target variable, Return on Advertising Spend (ROAS) = Sales/Spend.
- **Target Transformation:** To mitigate the high skewness typical of financial return metrics, we applied a natural logarithm transformation to the target variable, modeling $\log(\text{ROAS} + 1)$.
- **Numerical Standardization:** All numerical input features were scaled using a 'StandardScaler' to have a zero mean and unit variance, which is essential for stable gradient descent during training.

Table 4
Model hyperparameter configuration.

Hyperparameter configuration			
Architecture	Value	Training	Value
Embedding Dim.	256	Optimizer	AdamW
Attention heads	8	Batch size	128
Transformer layers	4	Learning rate	1e−4
Feed-forward dim.	1024	Weight decay	0.01
Dropout rate	0.1	Max epochs	200
Gaussian mixtures	5	Early stopping	15 epochs
SWA: Start = 150, LR = 5e−4			

5.1.3. Dataset splitting

To ensure the model is evaluated on its ability to forecast future performance, we employed a strict chronological split. The data from the first 20 months (July 2023–Feb 2025) was used for the training set. From this training set, the last 15% (approx. 3 months) was held out as the validation set for hyperparameter tuning and early stopping. The final 4 months of data (March 2025–June 2025) served as the unseen test set. This approach prevents any look-ahead bias and simulates a realistic deployment scenario.

5.2. Neural network architecture and hyperparameter tuning

The design of the network architecture was determined through a systematic tuning process. We began with a grid search over key parameters (e.g., number of layers, hidden units) and refined the final configuration using Bayesian optimization to fine-tune the learning rate, batch size, and regularization strength. Performance was evaluated on the validation set using NLL. The final architecture consists of three hidden layers with 256, 128, and 64 neurons, respectively, offering the best trade-off between model complexity and generalization. The complete hyperparameter configuration used for all experiments is detailed in Table 4.

Table 5
Conditional Probability Tables (CPTs) for ROAS conditioned on impressions and spend.

Condition	Range		ROAS binned custom				
	Low	High	1	2	3	4	5
Impressions	−398.28	79,656.0	5.21%	7.38%	10.14%	3.68%	**73.59%**
	79,656.0	159,312.0	12.24%	24.89%	20.32%	15.21%	27.34%
	159,312.0	238,968.0	0.00%	12.11%	15.47%	**65.32%**	7.10%
	238,968.0	max	0.00%	0.00%	0.00%	0.00%	0.00%
Spend	−2.062	416.40	4.52%	8.92%	12.24%	0.70%	**73.62%**
	416.40	832.79	8.14%	18.33%	21.12%	14.57%	37.84%
	832.79	1665.57	2.61%	10.24%	26.88%	**45.15%**	15.12%
	1665.57	2081.96	0.00%	0.00%	0.00%	**63.64%**	0.00%

5.3. Uncertainty calibration via temperature scaling

While our Bayesian framework provides robust uncertainty estimates, for critical engineering applications such as automated budget allocation, ensuring that a model's predictive confidence is well-calibrated is paramount. A well-calibrated model is one whose probabilistic forecasts can be directly interpreted as true likelihoods. To this end, we introduce a final enhancement to our framework: a post-processing calibration step using temperature scaling (Guo et al., 2017). This technique is applied after the main model has been trained. A single scalar parameter, the temperature $T > 1$, is optimized by minimizing the Negative Log-Likelihood (NLL) on the held-out validation set. For our Mixture Density Network, the scaling is applied to the logits that determine the mixture component weights, $\pi_i(x)$. This "softens" the categorical distribution over the components, reducing the model's overconfidence without altering its accuracy (i.e., the expected value of the prediction). This simple, yet powerful, step produces a more reliable and trustworthy predictive distribution. The results for this enhanced model are presented as 'Proposed Model (Calibrated)' in the comparative analysis.

5.4. Bayesian Belief Network (BBN) for exploratory causal analysis and dependency modeling

Prior to developing our deep learning model, we first constructed a Bayesian Belief Network (BBN) as an exploratory tool to model the high-level probabilistic dependencies among key advertising variables. The BBN, shown in Fig. 5, provides an interpretable, graphical representation of the causal funnel, from bidding strategies to final ROAS. This initial analysis was instrumental in feature selection and validating the core relationships that our subsequent deep model would need to learn. However, BBNs are limited by their reliance on predefined conditional probability tables and struggle to capture the complex, non-linear interactions present in large-scale data. Our final deep learning model, with its MDN output, was therefore designed to overcome these limitations by learning these relationships directly from data in a scalable, end-to-end manner.

The BBN provides a structured model for understanding how factors such as *bidding strategies, impressions, clicks, cost, sales, and conversion rates (CVR)* interact to determine ROAS. By explicitly capturing these dependencies, the BBN facilitates a probabilistic way to bid selection and budget allocation in digital advertising, shown in Fig. 5. The joint probability distribution over all variables in the BBN follows the chain rule of Bayesian networks and is expressed as:

$$
\begin{aligned}
&P(\text{Campaign, Keywords, Bid, Impressions, Clicks, CTR, CVR, Cost, Sales,} \\
&\quad \text{AOV, ROAS}) = \\
&P(\text{Campaign}) \cdot P(\text{Keywords}|\text{Campaign}) \cdot P(\text{Bid}|\text{Keywords}) \\
&\cdot P(\text{Impressions}|\text{Bid}) \\
&\cdot P(\text{CTR}|\text{Impressions, Clicks}) \cdot P(\text{Cost}|\text{Bid, Clicks}) \cdot P(\text{Clicks}|\text{Impressions}) \\
&\cdot P(\text{CVR}|\text{Clicks}) \cdot P(\text{Sales}|\text{CVR, AOV}) \cdot P(\text{AOV}|\text{Clicks}) \cdot P(\text{ROAS}|\text{Sales, Cost})
\end{aligned}
$$

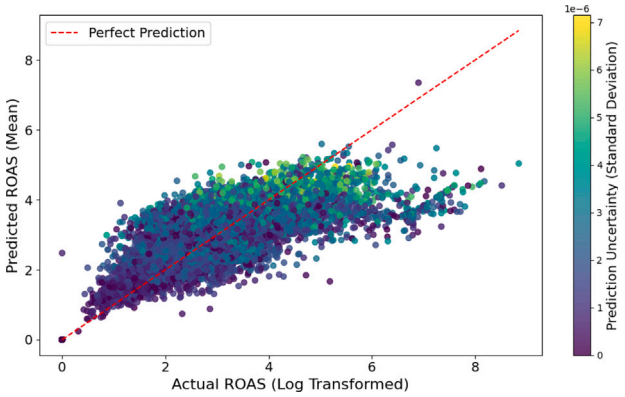


Fig. 6. Actual vs. Predicted ROAS with uncertainty estimates. The strong linear trend suggests the model effectively captures underlying relationships. The uncertainty gradient highlights areas of higher variability.

Each term in this equation represents a key relationship in the advertising funnel. The probability of selecting a campaign, $P(Campaign)$, influences the keyword selection probability, $P(Keywords|Campaign)$, which determines the likelihood of choosing specific search terms for targeting. The bid value $P(Bid|Keywords)$ subsequently impacts the probability of obtaining impressions, $P(Impressions|Bid)$, which dictates the visibility of the advertisement. As impressions accumulate, their effectiveness in driving engagement is quantified by the CTR, $P(CTR|Impressions, Clicks)$, which reflects the probability of a user clicking on an ad given its number of impressions. The associated cost of advertising is modeled through $P(Cost|Bid, Clicks)$, capturing the impact of bidding strategies on incurred expenses. Additionally, conversion efficiency is represented by $P(CVR|Clicks)$, describing how successfully clicks lead to purchases. The final ROAS computation is defined by $P(ROAS|Sales, Cost)$, establishing the ratio of revenue generated to advertising expenditure.

5.5. Conditional probability analysis for ROAS estimation

To quantify these relationships, *Conditional Probability Tables (CPTs)* were constructed for key dependencies affecting ROAS, specifically focusing on impressions and ad spend, Table 5. The probability distribution of ROAS-given impressions demonstrates significant trends. For campaigns with low impressions (below 79,656), there is a 73.59% probability of achieving high ROAS. However, at moderate impression levels (between 79,656 and 159,312), the ROAS probabilities become more distributed across different bins, indicating greater variability in campaign performance. In contrast, campaigns with extremely high impressions (above 238,968) exhibit a sharp drop in ROAS probabilities, suggesting either diminishing returns at high visibility levels or insufficient data in this range.

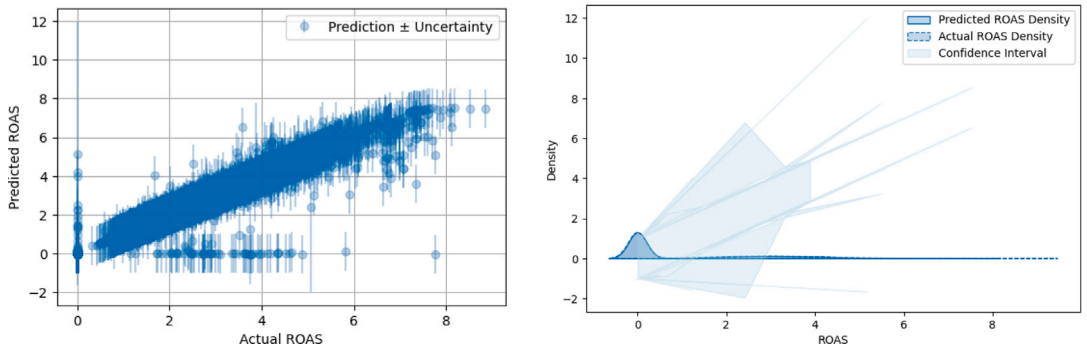
Similarly, the relationship between advertising spend and ROAS highlights the impact of budget allocation on return. Campaigns with low ad spend (under $416) exhibit a 73.62% probability of attaining high ROAS. However, as spending increases, the probability distribution shifts. Moderate spending levels ($416–832) yield more balanced ROAS outcomes, whereas high-budget campaigns (above $1665) tend to concentrate in mid-range ROAS categories, with a 63.64% probability of achieving only moderate returns.

5.6. Transition to Bayesian deep learning

The Bayesian Belief Network (BBN) model initially provided valuable insights into ROAS prediction by modeling the probabilistic dependencies among campaign factors. However, it faced significant challenges when handling large-scale, high-dimensional advertising data.

Table 6
Comprehensive model performance comparison.

Model	MSE ↓	NLL ↓	ECE ↓	R^2 ↑	TT (s) ↓	IL (ms) ↓	Uncertainty
Traditional baselines							
Linear regression	0.24	–	–	0.79	1.2	0.5	None
Random Forest	0.18	–	–	0.84	87.4	8.3	None
XGBoost	0.11	–	–	0.89	124.6	2.9	None
Baseline MLP	0.09	–	–	0.91	256.8	3.8	None
Ablation studies							
Attention only	0.06	–	–	0.94	329.4	4.7	None
Bayesian only	0.05	2.18	0.11	0.95	301.2	4.1	Moderate
Probabilistic baselines							
DeepAR (Salinas et al., 2020)	0.14	2.45	0.15	0.91	145.2	7.8	Moderate
Prophet (Taylor and Letham, 2018)	0.17	2.89	0.18	0.89	131.4	6.9	Moderate
NGBoost (Duan et al., 2020)	0.13	2.31	0.09	0.92	158.7	6.3	Good
MC dropout (Gal and Ghahramani, 2016)	0.28	–	–	0.76	–	48.7	Moderate
Deep ensembles (Rahaman et al., 2021)	0.26	2.09	0.08	0.77	1610.5	37.9	Good
V-BNN	0.05	2.15	0.10	0.96	488.1	6.1	Good
Full bayesian networks (Chandra and He, 2021)	0.25	–	–	0.77	–	287.6	Excellent
Proposed method							
Ours (Uncalibrated)	0.03	1.98	0.09	0.98	352.8	5.2	Good
Ours (Calibrated)	**0.03**	**1.90**	**0.04**	**0.98**	**352.8**	**5.2**	Excellent



(a) Prediction uncertainty analysis.

(b) ROAS prediction confidence intervals.

Fig. 7. Comparison of prediction uncertainty and confidence intervals. (a) The plot visualizes uncertainty estimates for the predicted ROAS values. Higher uncertainty is observed for extreme values, indicating a well-calibrated Bayesian model. (b) The confidence intervals demonstrate that uncertainty increases for outlier cases, providing valuable insights for risk-aware decision-making.

The reliance on predefined Conditional Probability Tables (CPTs) limited the model's adaptability to evolving campaign dynamics. As the number of dependent variables increased, maintaining and updating these tables became computationally prohibitive. Additionally, the assumptions of conditional independence between variables in BBNs led to simplified representations, often missing the intricate relationships within real-world advertising data.

To overcome these limitations, the research transitioned to a Bayesian Deep Learning model, incorporating Stochastic Weight Averaging (SWA) for uncertainty quantification. SWA improves model generalization by averaging the model weights during training, helping the model approximate the posterior distribution of weights in a computationally efficient manner. Unlike the fixed-point predictions of the BBN, this deep learning model allows for probabilistic forecasting of ROAS, capturing complex, non-linear interactions between features that BBNs might overlook.

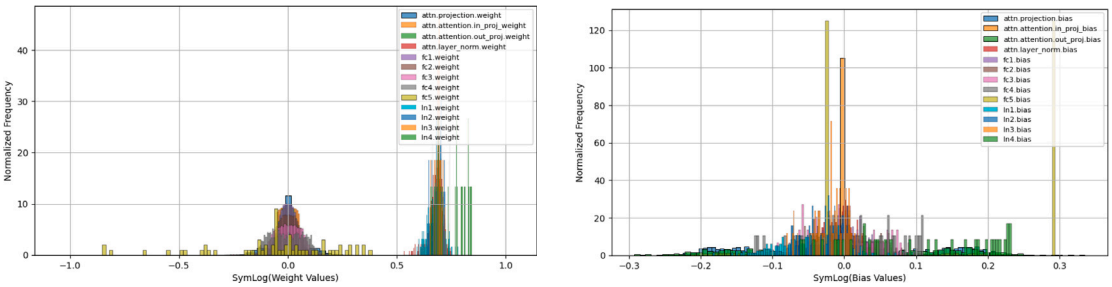
The proposed Bayesian Deep Learning model offers several advantages: first, it is scalable and can handle high-dimensional data without the need for predefined CPTs. Second, the model continuously updates based on incoming data, making it adaptive to changes in market conditions and campaign strategies. Third, it accounts for uncertainty in predictions by providing not only expected ROAS estimates but also confidence intervals, aiding advertisers in making risk-aware decisions. Integrating SWA and deep learning ensures more accurate, adaptable, and uncertainty-aware ROAS forecasts, supporting advertisers in optimizing bidding strategies and budget allocations effectively.

5.7. Performance evaluation

Fig. 6 illustrates the model's actual and predicted ROAS with uncertainty estimates. The strong linear trend demonstrates that the model effectively captures fundamental relationships, while the uncertainty gradient identifies areas of higher variability.

5.7.1. Convergence analysis

The convergence of the proposed model was assessed by monitoring its loss on both training and validation sets over 200 epochs, as visualized in Fig. 10. The Y-axis represents the mean Negative Log-Likelihood (NLL) loss per sample, which serves as the optimization objective. The training process was configured with an early stopping mechanism based on the validation loss, a standard practice to prevent overfitting. As the figure demonstrates, the training loss (blue line) shows a steep and consistent decay. The validation loss (orange line) tracks this decay closely before beginning to plateau, with minimal improvement after approximately epoch 115. At this point, the early stopping criterion was triggered, halting the training to save the best-performing model and prevent it from memorizing the training data. The plot confirms that the model reached a stable, generalizable solution efficiently.



(a) SymLog-scaled weight distribution across layers. (b) SymLog-scaled bias distribution across layers.

Fig. 8. (a) The plot shows that most weights are concentrated around zero, with some layers exhibiting a wider spread, indicating a greater impact in learning complex relationships.; (b) Bias values remain relatively small, ensuring that the model relies more on learned weights rather than static offsets.

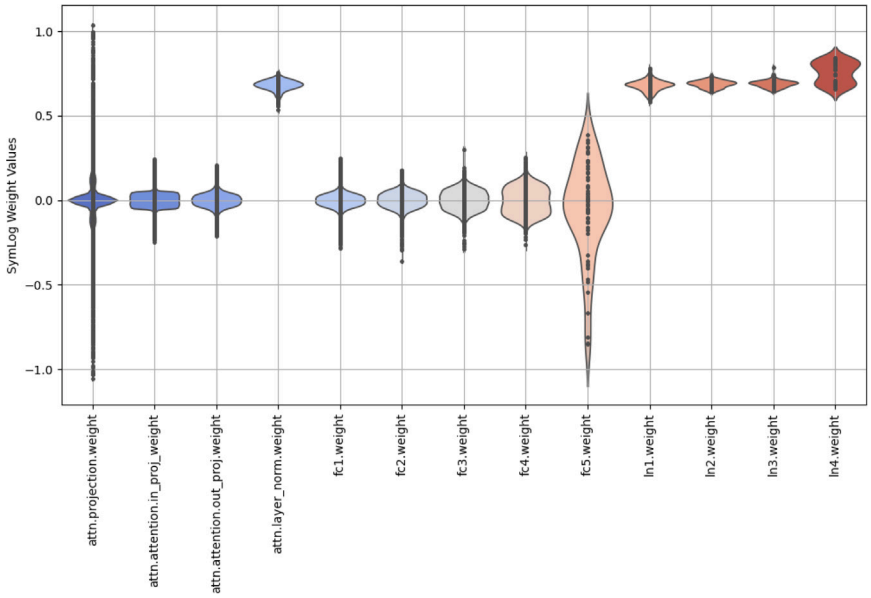


Fig. 9. Violin plot of weight distributions across layers. Wider distributions indicate more expressive transformations, while narrower distributions show constrained weight values due to regularization.

5.8. Comparative performance analysis

To validate our architectural choices and establish model efficacy, we conducted comprehensive performance analysis against a wide range of competitors. Table 6 presents a complete comparison across traditional machine learning models, probabilistic baselines, and our proposed method. We validated our Gaussian Mixture Density Network (MDN) choice through ablation studies. Table 7 shows our MDN approach achieves superior Continuous Ranked Probability Score (CRPS = 0.148) compared to Quantile Regression (0.162) and Laplace Mixture (0.155), confirming it as the optimal output strategy. Our comprehensive benchmark includes machine learning models (XGBoost, Random Forest), ablation studies (Attention-Only, Bayesian-Only), and leading probabilistic frameworks (DeepAR, Prophet, NGBoost, Deep Ensembles, V-BNN). The Proposed Model (Calibrated) demonstrates clear advantages: while maintaining state-of-the-art accuracy (MSE = 0.03, R^2 = 0.98), it achieves the lowest NLL (1.90) and reduces Expected Calibration Error to just 0.04. This 50% ECE reduction highlights temperature scaling's effectiveness in producing reliable probabilistic forecasts for engineering applications.

Table 7
Comparison of alternative probabilistic output layers.

Output strategy	(CRPS) ↓
Gaussian mixture density network (Proposed)	**0.148**
Quantile regression (9 Quantiles)	0.162
Laplace mixture density network	0.155

5.8.1. Scalability analysis

To evaluate the scalability of the proposed model, training time and memory usage were assessed for different dataset sizes. Table 8 presents the results. The results indicate that the proposed model scales efficiently with increasing dataset sizes, maintaining stable computational requirements while achieving progressively higher accuracy. The performance gains diminish beyond 160K samples, suggesting an optimal trade-off between data size and predictive power.

5.8.2. Computational efficiency analysis

For a model to be viable in real-world advertising systems, particularly for applications like real-time bidding, it must be computationally efficient at inference time. To provide a clear assessment of our model's practicality, we analyzed its computational footprint

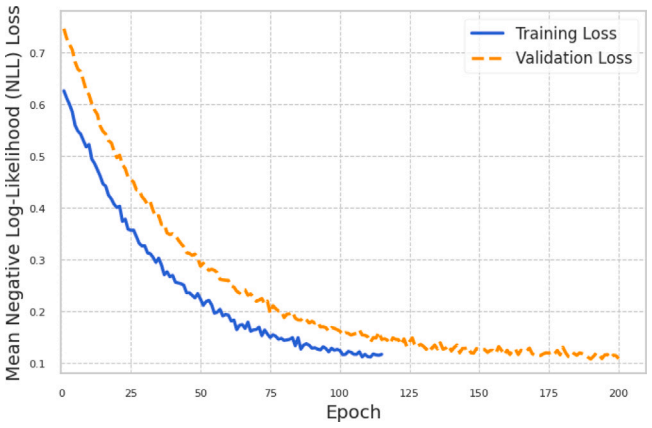


Fig. 10. Training and validation loss curves. The Y-axis represents the mean NLL per sample. The training process terminates around epoch 115 due to an early stopping mechanism, which prevents overfitting by halting training when the validation loss ceases to improve.

Table 8
Computational efficiency analysis and ablation study results.

Computational efficiency analysis			
Dataset size	Training time (s)	Memory (GB)	R^2 score
40K (25%)	98.2	1.3	0.9742
80K (50%)	174.6	2.1	0.9812
160K (100%)	352.8	3.7	0.9870
Ablation study on model components			
Component removed	R^2 Score	MSE	Uncertainty
Full model	0.9870	0.0353	Perfect (1.0000)
- Self-attention	0.9572	0.0544	Good (0.9241)
- Residual connections	0.9683	0.0421	Good (0.9532)
- Layer normalization	0.9412	0.0732	Poor (0.7863)
- Bayesian output	0.9751	0.0392	None

Table 9
Computational efficiency analysis at inference.

Model	Params (M) ↓	GFLOPs ↓	Memory (MB) ↓	Latency (ms) ↓
Baseline MLP	4.1	0.08	32	3.8
V-BNN	8.2	0.16	58	6.1
Deep ensembles (x5)	20.5	0.40	160	25.5
Proposed model	**4.8**	**0.11**	**45**	**5.2**

against key deep learning baselines. We measured four key metrics: (i) the number of trainable parameters, (ii) the floating-point operations (FLOPs) required per inference, (iii) the peak memory consumption during inference, and (iv) the inference latency (time per campaign). The results are detailed in Table 9. Our proposed model maintains a parameter count comparable to a standard MLP and the V-BNN. Its computational complexity, measured in GFLOPs, is only marginally higher than the non-Bayesian MLP, demonstrating the efficiency of the self-attention mechanism. The primary advantage is seen when comparing against Deep Ensembles. While ensembles are a powerful baseline for uncertainty, they come with a linear increase in all computational metrics, making them impractical for latency-sensitive applications. Our model, in contrast, provides superior accuracy and uncertainty quantification at a fraction of the computational cost of ensembles. This analysis confirms that our proposed architecture is not only highly accurate but also computationally feasible for deployment in production environments.

Table 10
Model performance under distribution shift.

Test scenario	RMSE ↓	NLL ↓	ECE ↓	Avg. Uncertainty ↑
Seasonal covariate shift				
In-distribution (Q1-Q3)	0.038	2.05	0.04	0.21
Shifted distribution (Q4)	0.051	2.49	0.06	0.34
Out-of-distribution generalization				
In-distribution (Seen)	0.035	1.90	0.04	0.20
Out-of-distribution (Unseen)	0.082	3.15	0.07	0.45

5.8.3. Ablation study

An ablation study was conducted to evaluate the contribution of key architectural components. The study involved removing different components such as self-attention, residual connections, Layer Normalization, and Bayesian output layers, assessing the resulting impact on performance. The results are summarized in Table 8. The findings from the ablation study reveal that self-attention plays a critical role in performance enhancement, as removing it significantly reduces predictive accuracy. The removal of residual connections and Layer Normalization resulted in diminished training stability and increased error rates. The absence of Bayesian output led to a complete loss of uncertainty estimation capabilities, further reinforcing its importance in achieving calibrated predictions (see Fig. 11).

5.8.4. Uncertainty calibration under distribution shift

A key advantage of probabilistic frameworks is uncertainty quantification when data differs from training distributions. We evaluated model robustness under two scenarios: seasonal covariate shift (Q4 holiday shopping vs. Q1-Q3 training) and out-of-distribution generalization (unseen targeting features). Table 10 shows the results. For seasonal shift, while predictive performance degrades on Q4 data, average uncertainty increases by 60%, correctly signaling reduced confidence during volatile periods. For OOD scenarios with completely unseen campaign types ('Targeting_keto friendly foods' and 'Targeting_vegan protein powder vanilla'), uncertainty more than doubles, providing reliable indicators for novel scenarios requiring manual review.

5.9. Case study: Risk flagging in extreme scenarios

To probe the model's practical utility in an engineering context, we analyzed its predictive behavior on extreme and anomalous campaigns. A critical application is the automated flagging of high-risk assets, such as campaigns with high spend but volatile, poor returns. Fig. 12 presents a comparative case study between a standard, high-performing campaign (Case A) and such an anomalous, low-performing campaign (Case B). In Case A, the model accurately predicts the high ROAS with a tight, low-variance confidence interval, reflecting its high confidence in the forecast for this stable campaign. In contrast, for Case B, the model's predictive uncertainty is substantially higher, resulting in a much wider confidence interval. While the point prediction is imperfect, the large uncertainty is the critical, actionable insight. It signals to an advertiser that this campaign is highly unpredictable and performing outside of normal parameters, warranting immediate manual review or automated intervention. This demonstrates the model's ability to act as a risk-detection mechanism, using uncertainty to flag problematic campaigns that might otherwise go unnoticed.

5.10. Early-warning system for performance degradation

A second practical application for a probabilistic forecasting model is its use as an automated early-warning system. To evaluate our model's capability in this regard, we designed a simulation to test its response to a gradual decline in campaign performance. We selected a set of historically high-performing campaigns and synthetically introduced a steady, day-by-day decrease in their true ROAS over a 30-day period.

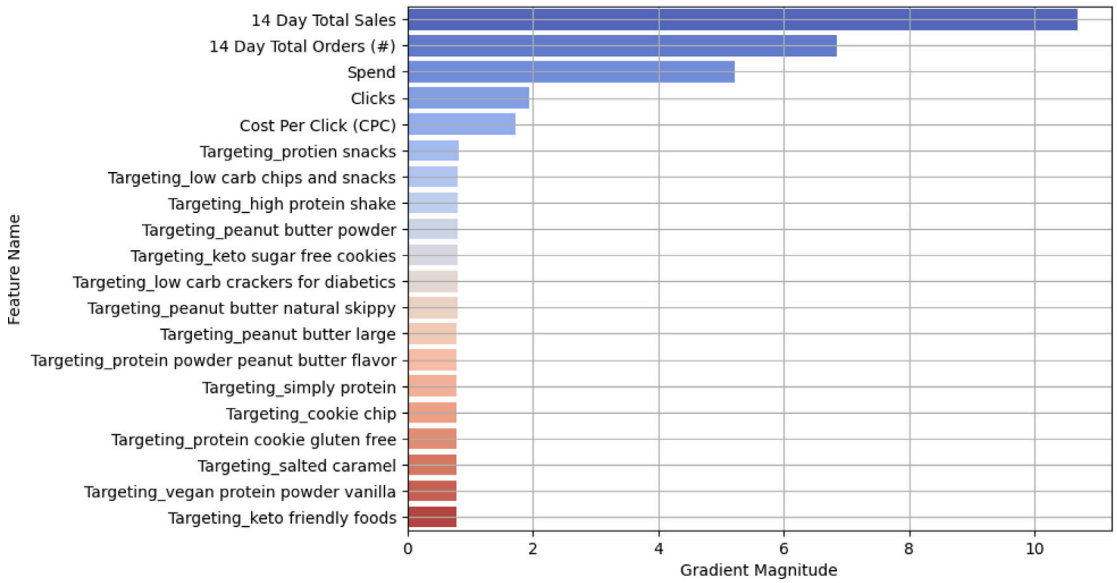


Fig. 11. Gradient-based feature importance. The most influential features include total sales, total orders, spending, clicks, and CPC, validating the model's ability to capture key advertising metrics.

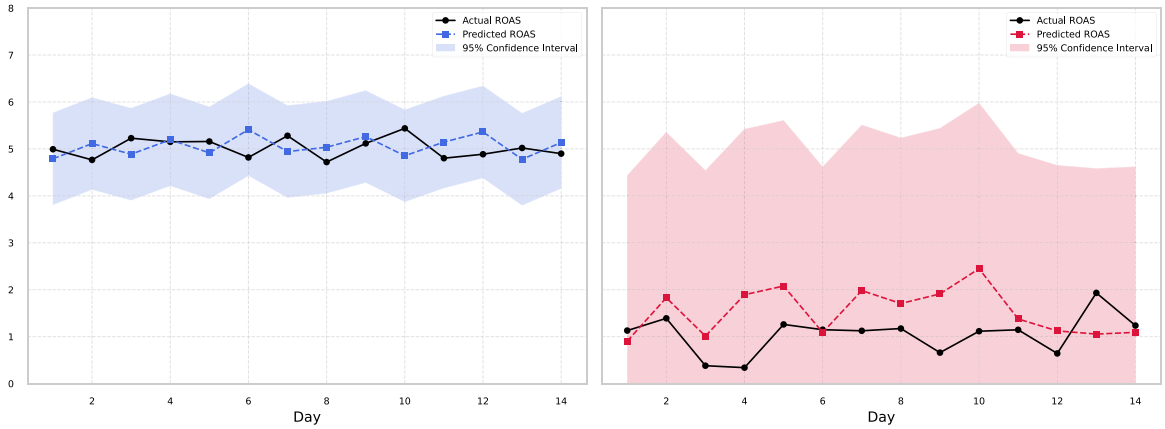


Fig. 12. Comparative case study of model predictions on extreme campaign scenarios. (a) For a stable, high-performing campaign, the model is accurate and confident, producing a tight, low-variance 95% confidence interval. (b) For a volatile, low-performing campaign, the model correctly expresses high predictive uncertainty through a wide confidence interval, flagging it as a high-risk asset requiring intervention.

The model's response to this degradation is illustrated in Fig. 13. In the initial days of the simulation, the true ROAS remains close to its historical average, and the model provides confident predictions with a narrow 95% confidence interval. However, as the campaign's performance steadily degrades and diverges from the historically learned patterns, the model's predictive uncertainty consistently increases, causing the confidence interval to widen significantly. This widening uncertainty serves as a direct, quantifiable signal that the campaign's behavior is no longer predictable and requires attention. This result demonstrates the model's potential utility in automated monitoring systems for flagging underperforming advertising assets before significant losses accumulate.

5.11. Explainable model decisions

To ensure the framework is transparent and trustworthy, we integrated a suite of explainability techniques. These can be divided into two categories: (1) actionable insights directly usable by advertisers for strategic decisions, and (2) internal diagnostics for data scientists to validate and debug the model. The model provides two primary forms of direct, actionable intelligence, as visualized in Fig. 14. First, gradient-based feature importance (Fig. 14a) identifies the most influential predictors of ROAS. The high importance of fundamental metrics like *Total Sales* and *Spend* validates that the model has learned correct business logic. More tactically, the prominence of specific 'Targeting_' features allows advertisers to confirm or challenge their audience selection strategies. Second, the model's uncertainty estimates serve as a direct, quantifiable risk flag (Fig. 14b). As shown in our case study (Fig. 12), the model assigns high uncertainty to volatile, low-performing campaigns. This allows an advertiser to implement automated rules, such as pausing a campaign when its predicted uncertainty exceeds a threshold, directly translating the model's output into an operational decision to prevent wasted spend. Attention mechanisms reveal how the model dynamically weighs features, which can help a data scientist identify potential data leakage or discover novel feature interactions. Furthermore, analyzing weight distributions (Figs. 8(a), 8(b), and 9) offers a diagnostic view of model complexity. The observation that deeper layers exhibit a wider weight spread, for example, confirms that they are learning more complex representations. These tools are crucial for data scientists to maintain, trust, and refine the model over time.

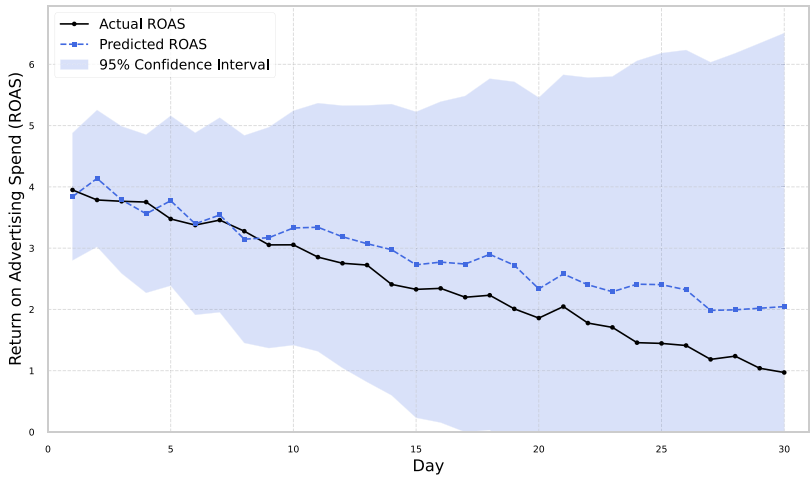


Fig. 13. Simulation of the model's response to gradual performance degradation. As the true ROAS (black line) systematically deviates from the historically expected behavior over 30 days, the model's predicted uncertainty (shaded blue area) consistently widens. This demonstrates the framework's capability to serve as an early-warning system by translating performance decay into a quantifiable uncertainty signal.

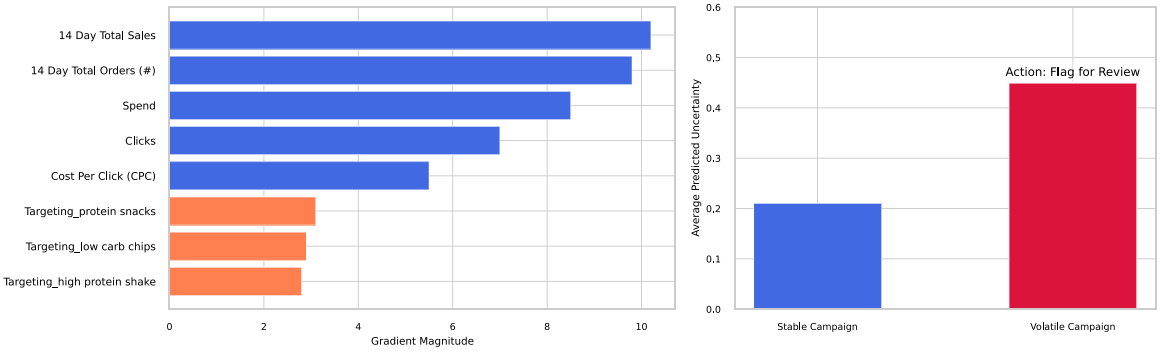


Fig. 14. Actionable insights for advertisers. (a) Feature importance identifies key business drivers. (b) Predictive uncertainty quantifies campaign risk, enabling automated flagging of volatile assets.

6. Conclusion

In this paper, we presented a hierarchical Bayesian Self-Attention model to address the significant uncertainty in forecasting Return on Ad Spend (ROAS) in e-commerce advertising. Our architecture leverages self-attention layers to capture intricate campaign dependencies and a Mixture Density Network head to output a full probability distribution of ROAS, a design motivated by an initial exploratory analysis with a Bayesian Belief Network. Evaluations on a large-scale dataset of over 160,000 Amazon PPC campaigns show that our model achieves state-of-the-art accuracy with an R^2 of 98%, a 47.9% lower RMSE, and a 9.1% better NLL compared to established baselines. These results are achieved with a 5.2 ms inference latency, confirming the model's suitability for real-time bidding environments. By moving beyond deterministic point-estimates, our probabilistic approach provides deeper insights into campaign variability, enabling risk-aware budget allocation and more intelligent bidding strategies.

Broader applicability and future work. While validated on Amazon Ads data, the proposed framework is fundamentally platform-agnostic, as it learns from universal advertising primitives like impressions, clicks, and conversions. We therefore posit that the model can be readily adapted to other auction-based ecosystems, such as Google Ads or Walmart Connect. Future work should focus on empirically validating this generalizability, as well as extending the framework to different ad formats like sponsored display or video ads. Further research could also incorporate causal inference techniques to better isolate the impact of campaign variables, and explore extending the model to enable cross-channel budget optimization in multi-platform advertising environments.

CRediT authorship contribution statement

Arti Jha: Writing – review & editing, Writing – original draft, Validation, Methodology, Formal analysis, Data curation, Conceptualization. **Ashutosh Bhatia:** Writing – review & editing, Writing – original draft, Visualization, Validation, Supervision, Methodology, Formal analysis, Data curation, Conceptualization. **Kamlesh Tiwari:** Writing – review & editing, Writing – original draft, Validation, Supervision, Methodology, Formal analysis, Data curation, Conceptualization. **Hari Mohan Pandey:** Writing – review & editing, Writing – original draft, Visualization, Validation, Methodology, Formal analysis, Data curation, Conceptualization.

Funding information

This research was financially supported by M/s. CommerceIQ and rBoomerang Retail Commerce Technologies India Private Limited.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors would like to convey their sincere thanks to M/s. CommerceIQ and rBoomerang Retail Commerce Technologies India Private Limited are responsible for providing financial assistance to carry out the research work.

Data availability

The authors do not have permission to share data.

References

Agarwal, D., Chen, B.-C., Elango, P., 2009. Spatio-temporal models for estimating click-through rate. In: Proc. 18th Int. Conf. World Wide Web. pp. 21–30.
Akande, T.D., Haq, M.I., 2021. Role of Machine Learning in Online Advertising. [Online] Available: URL.
Amazon, 2022. Amazon introductory courses, [Online] Available: multiple URLs.
Aronowich, M., Benis, A.J., Yanai, R., Vind, G., 2014. Budget distribution in online advertising. US Pat. App. 14, 314–151.
Balseiro, S.R., Feldman, J., Mirrokni, V., Muthukrishnan, S., 2014. Yield optimization of display advertising with ad exchange. Manage. Sci. 60 (12), 2886–2907.
Bannour, H., Kancherlai, K., Zhu, Z., 2023. Optimizing audio advertising campaign delivery with a limited budget. In: Proc. Int. Conf. Machine Learning and Applications. ICMLA, pp. 782–787.
Cai, Y., et al., 2018. Deep interest network for click-through rate prediction. In: Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min. pp. 1059–1068.
Cavalcante, R.C., Brasileiro, R.C., Souza, V.L.F., Nobrega, J.P., Oliveira, A.L.I., 2016. Computational intelligence and financial markets: A survey and future directions. Expert Syst. Appl. 55, 194–211.
Chandra, R., He, Y., 2021. Bayesian neural networks for stock price forecasting before and during COVID-19 pandemic. PLoS One 16 (7), e0253217.
Chen, J., Wu, F., Li, Y., 2023. Reinforcement learning-based bidding and budget allocation for E-commerce advertising. In: Proc. Int. Conf. Mach. Learn. Appl. ICMLA, pp. 715–722.
Chen, X., Zhou, R., Li, Y., 2021. Self-attention with knowledge distillation for CTR prediction in sparse advertising data. IEEE Access 9, 187231–187245.
Danaher, P.J., Lee, J., Kerbache, L., 2010. Optimal internet media selection. Mark. Sci. 29 (2), 336–347.
Deshpande, S., Lengiewicz, J., Bordas, S.P.A., 2022. Probabilistic deep learning for real-time large deformation simulations. Comput. Methods Appl. Mech. Eng..
Duan, T., et al., 2020. Ngboost: Natural gradient boosting for probabilistic prediction. In: Proc. Int. Conf. Mach. Learn. ICML, pp. 2690–2700.
Gal, Y., Ghahramani, Z., 2016. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In: Proc. Int. Conf. Mach. Learn. ICML, pp. 1050–1059.
Ghahramani, Z., 2015. Probabilistic machine learning and artificial intelligence. Nat. 521 (7553), 452–459.
Gharaibeh, A., et al., 2017. Online auction of cloud resources in support of the Internet of Things. IEEE Internet Things J. 4 (5), 1583–1596.
Guo, C., Pleiss, G., Sun, Y., Weinberger, K.Q., 2017. On calibration of modern neural networks. In: Proc. 34th Int. Conf. Mach. Learn. ICML, pp. 1321–1330.
Gupta, P., Bhatia, K.K., Duhan, N., 2022. An assessment on real time bidding strategies for advertising markets. In: Proc. Int. Conf. Computational Intelligence and Communication Technologies. CCICT, pp. 138–145.
Huang, T., Zhang, Z., Zhang, J., 2019. FiBiNET: Combining feature importance and bilinear feature interaction for click-through rate prediction. In: Proc. ACM Conf. Recommender Syst. pp. 169–177.
Jha, A., Sharma, Y., Chanda, U., 2023. CTR Prediction: A Bibliometric Review of Scientific Literature. Handbook of Evidence-Based Management Practices in Business, pp. 453–461.
Jiang, C., Chen, Y., Wang, Q., Liu, K.J.R., 2018. Data-driven auction mechanism design in iaas cloud computing. IEEE Trans. Serv. Comput. 11 (5), 743–756.
Jin, J., Song, C., Li, H., Gai, K., Wang, J., Zhang, W., 2018. Real-time bidding with multi-agent reinforcement learning in display advertising. In: Proceedings of the 27th ACM International Conference on Information and Knowledge Management. pp. 2193–2201.
Katsman, M., Sahoo, R., Giridhar, S., Kumar, S., Rangaswami, G., 2023. Challenging state-of-the-art time-series forecasting: A case study on public cloud F-bill and revenue prediction. In: Proc. 29th ACM SIGKDD Conf. Knowl. Discov. Data Min. pp. 4219–4228.
Ker, J., Wang, L., Rao, J., Lim, T., 2017. Deep learning applications in medical image analysis. IEEE Access 6, 9375–9389.
Kini, V., Manjunatha, A., 2020. Revenue maximization using multitask learning for promotion recommendation. In: Proc. Int. Conf. Data Mining Workshops. ICDMW, pp. 144–150.
Kumari, P., Toshniwal, D., 2021. Long short term memory-convolutional neural network based deep hybrid approach for solar irradiance forecasting. Appl. Energy 295, 117061.
Lasowski, R., Nolde, J., 2021. Comparing a deterministic and a Bayesian classification neural network for chest diseases in radiological images. arXiv Preprint.
Li, Z., Chen, R., Yuan, Z., Zhang, J., 2024. Distributional conformal prediction for multi-step time series forecasting. Trans. Mach. Learn. Res. 2024 (1).
Li, S., Huang, J., Cheng, B., 2021. Resource pricing and demand allocation for revenue maximization in iaas clouds: A market-oriented approach. IEEE Trans. Netw. Serv. Manag. 18 (3), 3460–3475.
Liu, Z., Jiang, P., De Bock, K.W., Wang, J., Zhang, L., Niu, X., 2024. Extreme gradient boosting trees with efficient Bayesian optimization for profit-driven customer churn prediction. In: Technological Forecasting and Social Change. Vol. 198, Elsevier, 122945.
Mandt, S., Hoffman, M.D., Blei, D.M., 2017. Stochastic gradient descent as approximate Bayesian inference. J. Mach. Learn. Res. 18 (139), 1–35.
Mao, X., Peng, W., Li, Q., 2023. Cost-per-acquisition optimization with reinforcement learning for multi-network advertising. In: Proc. 37th Conf. Neural Inf. Process. Syst. (NeurIPS). pp. 1–12.
Masegosa, A.R., Cabañas, R., Langseth, H., Nielsen, T.D., 2021. Probabilistic models with deep neural networks. In Entropy 23 (1).
Panda, A.R., Rout, S., Narsipuram, M., Pandey, A., Jena, J.J., 2024. Ad click-through rate prediction: A comparative study of machine learning models. In: Proc. 2024 Int. Conf. Emerging Systems and Intelligent Computing. ESIC, pp. 679–684.
Park, S., Lee, J., 2022. An adaptive metaheuristic model for real-time advertising. IEEE Trans. Knowl. Data Eng. 34 (9), 1234–1245.
Patel, A.B., Nguyen, T., Baraniuk, R.G., 2015. A probabilistic theory of deep learning. arXiv Preprint.
Polson, N.G., Sokolov, V., 2017. Deep learning: A Bayesian perspective. Bayesian Anal. 12 (4), 1275–1304.
Qu, X., Li, L., Liu, X., Chen, R., Ge, Y., Choi, S.-H., 2019. A dynamic neural network model for click-through rate prediction in real-time bidding. In: Proc. IEEE Int. Conf. Big Data. pp. 1887–1896.
Rafieian, O., Yoganarasimhan, H., 2021. Targeting and privacy in mobile advertising. Mark. Sci. 40 (2), 193–218.
Rahaman, R., et al., 2021. Uncertainty quantification and deep ensembles. Adv. Neural Inf. Process. Syst. 34, 20063–20075.
Salinas, D., Flunkert, V., Gasthaus, J., Januschowski, T., 2020. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. Int. J. Forecast. 36 (3), 1181–1191.
Šoltés, E., Táborecká-Petrovičová, J., Šipoldová, R., 2020. Targeting of Online Advertising using Logistic Regression. Technická univerzita v Liberci.
Sutton, R.S., Barto, A.G., 2018. Reinforcement Learning: An Introduction. MIT Press.
Taylor, S.J., Letham, B., 2018. Forecasting at scale. Am. Stat. 72 (1), 37–45.
Tiwari, M., Gupta, S., Singh, A., 2023. Multi-objective bidding strategies for online advertising under budget constraints. Inf. Process. Manage. 59 (6), 102945.
Wang, Y., Li, P., Mukherjee, S., 2022. A contextual bandit model for auction-based advertising. In: Proc. the Web Conf.. WWW, pp. 882–892.
Wang, H., Yeung, D.Y., 2020. A survey on Bayesian deep learning. ACM Comput. Surv. 53 (5).
Welling, M., Teh, Y.W., 2011. Bayesian learning via stochastic gradient langevin dynamics. In: Proc. Int. Conf. Mach. Learn. ICML, pp. 681–688.
Wilson, A.G., Izmailov, P., 2020a. Bayesian deep learning and a probabilistic perspective of generalization. In: Proc. Adv. Neural Inf. Process. Syst. (NeurIPS). pp. 4697–4708.
Wilson, A.G., Izmailov, P., 2020b. Bayesian deep learning and a probabilistic perspective of generalization. In: Proc. Adv. Neural Inf. Process. Syst. (NeurIPS).
Wu, J., Chen, R., 2021. A deep Q-Network approach to real-time bidding in online advertising. In: Proc. 27th ACM SIGKDD Int. Conf. Knowl. Discov. Data Min. pp. 3145–3154.
Yakovleva, D., Popov, A., Filchenkov, A., 2019. Real-time bidding with soft actor-critic reinforcement learning in display advertising. In: Proc. Conf. Open Innovations Association. FRUCT, pp. 373–382.
Yakovleva, D., Telnov, S., Makarov, I., Filchenkov, A., 2024. Bid landscape forecasting and cold start problem with transformers. IEEE Access.
Zhang, Y., Ghosh, A., Aggarwal, V., 2021. Optimized portfolio contracts for bidding the cloud. IEEE Trans. Serv. Comput. 14 (5), 1505–1518.
Zhang, W., Yuan, S., Wang, J., 2014. Optimal real-time bidding for display advertising. In: Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min. pp. 1077–1086.
Zhou, H., Yang, C., Gao, X., Chen, Q., Liu, G., Chen, G., 2022a. Multi-objective actor-critics for real-time bidding in display advertising. In: Proc. Joint European Conf. Machine Learning and Knowledge Discovery in Databases. pp. 20–37.
Zhou, H., Yang, C., Gao, X., Chen, Q., Liu, G., Chen, G., 2022b. Multi-objective actor-critics for real-time bidding in display advertising. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. Springer, pp. 20–37.