

Personalized Interpolation: Achieving Efficient Conversion Estimation with Flexible Optimization Windows

Xin Zhang
xinzhang5@meta.com
Meta Platforms, Inc.
USA

Weiliang Li
weiliangli@meta.com
Meta Platforms, Inc.
USA

Rui Li
ruli23@meta.com
Meta Platforms, Inc.
USA

Zihang Fu
fuzihang@meta.com
Meta Platforms, Inc.
USA

Tongyi Tang
tongyitang@meta.com
Meta Platforms, Inc.
USA

Zhengyu Zhang
zhengyuzhang@meta.com
Meta Platforms, Inc.
USA

Wen-Yen Chen
wychen@meta.com
Meta Platforms, Inc.
USA

Nima Noorshams
nshams@meta.com
Meta Platforms, Inc.
USA

Nirav Jasapara
niravj@meta.com
Meta Platforms, Inc.
USA

Xiaowen Ding
xwding@meta.com
Meta Platforms, Inc.
USA

Ellie Wen
ellie.wen@meta.com
Meta Platforms, Inc.
USA

Xue Feng
xfeng@meta.com
Meta Platforms, Inc.
USA

Abstract

Optimizing conversions is crucial in modern online advertising systems, enabling advertisers to deliver relevant products to users and drive business outcomes. However, accurately predicting conversion events remains challenging due to variable time delays between user interactions (e.g., impressions or clicks) and the actual conversions. These delays vary substantially across advertisers and products, necessitating flexible optimization windows tailored to specific conversion behaviors. To address this, we propose a novel *Personalized Interpolation* method that extends existing models based on fixed conversion windows to support flexible advertiser-specific optimization windows. Our method enables accurate conversion estimation across diverse delay distributions without increasing system complexity. We evaluate the effectiveness of the proposed approach through extensive experiments using a real-world ads conversion model. Our results show that this method achieves both high prediction accuracy and improved efficiency compared to existing solutions. This study demonstrates the potential of our Personalized Interpolation method to improve conversion optimization and support a wider range of advertising strategies in large-scale online advertising systems.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
CIKM '25, Coex, Seoul, Korea

© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-XXXX-X/2025/11
<https://doi.org/XXXXXXXX.XXXXXXX>

CCS Concepts

- **Computing methodologies** → **Machine learning algorithms;**
- **Information systems** → **Online advertising.**

Keywords

Ads Conversion Model, Flexible Optimization Window, Personalized Interpolation, Data Delay

ACM Reference Format:

Xin Zhang, Weiliang Li, Rui Li, Zihang Fu, Tongyi Tang, Zhengyu Zhang, Wen-Yen Chen, Nima Noorshams, Nirav Jasapara, Xiaowen Ding, Ellie Wen, and Xue Feng. 2025. Personalized Interpolation: Achieving Efficient Conversion Estimation with Flexible Optimization Windows. In *Proceedings of The 34th ACM International Conference on Information and Knowledge Management (CIKM '25)*. ACM, New York, NY, USA, 8 pages. <https://doi.org/XXXXXXXX.XXXXXXX>

1 Introduction

Data freshness plays a pivotal role in the efficacy of online advertising recommendation systems [17, 18]. The precision of predicting user conversions depends heavily on how promptly users' latest interests are reflected in both labels and features. Over the past decades, the advertising industry has made substantial strides in increasing model training frequency, evolving from daily updates to hourly and even real-time processing. Despite these advancements, a significant challenge remains on the advertiser's side, particularly regarding their preference to optimize for conversions that may occur long after a user is exposed to an ad. This duration, known as the *optimization window*, is crucial as it determines the metrics by which ad performance is evaluated.

Long optimization windows pose a formidable challenge to ad delivery systems, since predictive models must endure lengthy delays to accumulate the complete label distributions needed to align

with advertisers’ true objectives. Such delays can severely degrade performance; for example, a model that waits seven days to train on complete data may become obsolete for predicting conversions within that period. To address challenges associated with data delays, a new research domain — delayed feedback modeling (DFM) — has emerged [3, 16, 29]. DFM employs Bayesian inference and innovative architectures to balance data freshness and completeness, enabling optimization for fixed long windows.

Despite advancements in managing fixed long optimization windows, significant gaps remain in accommodating advertisers’ precise needs. The length of the optimization window often varies depending on the nature of the advertiser’s products. For instance, gaming advertisers targeting new users may face longer cycles from app installation to specific in-app events, with this duration varying by game type and target audience. These advertisers often have deeper insights into appropriate conversion windows for their products. Similarly, e-commerce advertisers seeking to boost purchases may operate within different time frames influenced by promotional periods prior to counting conversions. These diverse requirements create unique opportunities for ad companies to develop machine learning solutions that align more closely with advertiser goals.

Ideally, recommendation systems should be versatile enough to support any optimization window specified by advertisers. However, our literature review reveals a *lack of* prior research addressing this capability despite its practical importance. Developing a *flexible optimization window (FOW)* ad delivery system introduces new challenges in both modeling and infrastructure. Beyond managing data delays, scalability becomes a bottleneck, requiring resource-efficient solutions capable of supporting varied window scenarios.

In this paper, we develop an efficient and scalable method for FOW estimation that requires minimal system resources. Our major contributions are as follows:

- We conceptualize flexible optimization window estimation as approximating the cumulative distribution function (CDF) of conversion event times. Leveraging the monotonic increasing and concave properties of the CDF, we propose a novel personalized interpolation method to efficiently estimate conversions with flexible optimization windows. Our approach integrates seamlessly with existing production black-box models without incurring additional computational cost or system overhead.
- Our personalized interpolation method introduces a key parameter, the *interpolation factor*, which governs how conversion estimates are adjusted across different optimization windows. Since this factor is not directly observable, we propose three distinct parametric functions to approximate it, each offering different trade-offs in terms of complexity, interpretability, and performance. To better understand their effectiveness, we conduct an ablation study to analyze and compare their performance.
- To evaluate the effectiveness of our proposed method, we perform extensive experiments comparing it against both the theoretical upper bound and practical baselines suitable for production. Results indicate that our method not only achieves efficient FOW estimation but also delivers prediction accuracy close to the theoretical optimum — without requiring additional training or infrastructure resources.

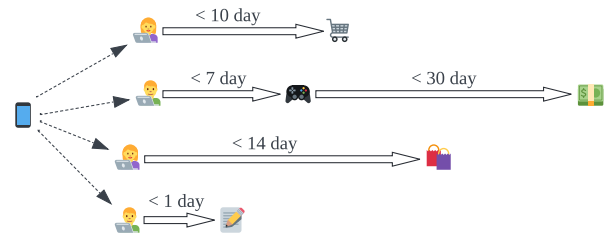


Figure 1: Illustration of varying conversion cycles from user interaction with ad impressions to final conversion events. The length of these cycles differs depending on the product type or conversion goal (e.g., form submission, game level achievement, or purchase). As a result, advertisers are motivated to optimize for different conversion windows tailored to their specific objectives.

The rest of this paper is organized as follows. Section 2 provides preliminaries and reviews related work. Section 3 formally formulates the FOW problem based on the long optimization window framework and highlights key challenges from modeling and system perspectives. Section 4 details our proposed Personalized Interpolation method and its integration with production black-box models. Section 5 presents experimental results, and Section 6 concludes the paper.

2 Related Works

Delayed Feedback Modeling for Conversion Prediction. Modeling conversion delays has been extensively studied in online advertising. Early work focused on explicitly modeling the time between impressions or clicks and conversions. Chapelle (2014) [3] and Ji et al. (2016) [13] proposed parametric models using exponential and Weibull distributions, respectively, though these often fail to capture complex delay patterns due to strong distributional assumptions.

To improve flexibility, Yoshikawa et al. (2018) [30] applied kernel density estimation (KDE). More recently, Wang et al. (2020) [24] jointly modeled click-through rate (CTR), conversion rate (CVR), and conversion delay by discretizing conversion time and applying softmax classification.

Another research direction addresses delayed feedback via label correction and reweighting. FSIW [29] uses importance sampling to adjust loss weights based on elapsed time. Extensions such as nnDF and DEFUSE [4, 28] provide unbiased CVR estimation, while UCL [23] jointly trains CVR and label correction models. ES-DFM [27] dynamically samples cut-off times to balance data freshness and label completeness. Several of these methods [16, 27] have shown success in industrial settings.

However, most approaches require dedicated delay models or complex training pipelines, increasing deployment costs. In contrast, our work reuses outputs from existing models to estimate conversion rates at arbitrary elapsed times—offering a more practical and efficient solution.

Survival Analysis and Interpolation Method. FOW estimation can be framed as approximating the cumulative distribution function (CDF) of conversion time, similar to problems in survival analysis [1, 7, 12, 22]. Survival models, widely used in healthcare

and engineering, estimate the survival function $S(t) = \mathbb{P}(T > t)$, where T denotes event time.

Classical approaches include the Kaplan–Meier estimator [14], Cox proportional hazards model [2, 5], and accelerated failure time model [25]. Recent deep learning variants such as DeepSurv [15] and DASA [21] improve performance but require intensive computation and model design, limiting industrial scalability.

Interpolation techniques, widely used in statistics and numerical analysis [6, 8], offer an efficient alternative for estimating values between known data points. Prior works [9, 20, 26] have combined interpolation with survival models to handle large or incomplete datasets.

Building on this idea, our approach uses interpolation over predictions from existing models to efficiently estimate conversion rates for arbitrary time windows—offering both accuracy and ease of deployment.

3 Problem Statement

In this section, we first describe the ads conversion model designed for long optimization window scenarios. Building on this formulation, we then extend the framework to address the Flexible Optimization Window (FOW) problem, and discuss the key challenges associated with enabling flexibility in both modeling and system implementation.

3.1 Preliminary for Long Optimization Window

We define the probability of an ad conversion event e occurring within a given optimization window T as $\mathbb{P}(\tau \in (0, T] | e)$, where τ denotes the time elapsed from a user’s click on the ad to the actual conversion. The value of T represents the optimization window, i.e., the maximum delay allowed for a conversion to be attributed to the ad click.

For short optimization windows (e.g., $T \leq 1$ day), the model can typically be trained directly, as the performance is relatively robust to short delays in label availability. However, when the optimization window becomes long (e.g., $T = 7$ days), delayed feedback becomes a critical bottleneck, significantly impacting model training and prediction accuracy.

A common approach to mitigate this issue is to decompose the long optimization window conversion probability into more manageable sub-problems - particularly by leveraging predictions from a short window where data is fresher and more accurate. Specifically, the conversion probability over a long window can be decomposed as follows:

$$\mathbb{P}(\tau \in (0, T_l] | e) = \mathbb{P}(\tau \in (0, T_s] | e) + \mathbb{P}(\tau \in (T_s, T_l] | e) \quad (1)$$

$$= \mathbb{P}(\tau \in (0, T_s] | e) + \left(1 - \mathbb{P}(\tau \in (0, T_s] | e)\right) \cdot \mathbb{P}(\tau \in (T_s, T_l] | \tau > T_s, e) \quad (2)$$

Here, T_s and T_l denote the short and long optimization windows, respectively. This formulation decomposes the long window prediction into two components: the probability of conversion within the short window $(0, T_s]$, and the conditional probability of converting in the remaining window $(T_s, T_l]$, given that the conversion did not occur before T_s . Equations (1) and (2) represent two alternative decompositions. Notably, (2) explicitly conditions the second term on the event $\tau > T_s$, thereby making more effective use of the

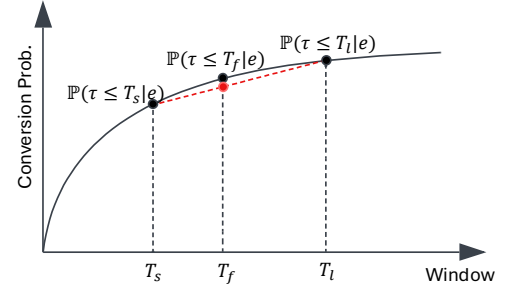


Figure 2: Illustration of the interpolation framework on the conversion CDF. Given the conversion probabilities $\mathbb{P}(\tau \leq T_s | e)$ and $\mathbb{P}(\tau \leq T_l | e)$, the probability at an intermediate window $\mathbb{P}(\tau \leq T_f | e)$ (black dot) can be approximated via a linear combination of the two endpoints (red dot).

short-window prediction $\mathbb{P}(\tau \in (0, T_s] | e)$, which is typically more accurate due to fresher data availability.

3.2 Problem Formulation of FOW and its Challenges

Building on the long optimization window framework, we can reuse the formulation from (1) or (2) by replacing the long window T_l with the desired window T_f . While the short-window prediction $\mathbb{P}(\tau \in (0, T_s] | e)$ can still be leveraged, we now require an additional model to estimate either $\mathbb{P}(\tau \in (T_s, T_f] | e)$ or the conditional probability $\mathbb{P}(\tau \in (T_s, T_f] | \tau > T_s, e)$.

However, this approach quickly becomes impractical when scaling to support multiple advertisers with varying optimization windows. For example, supporting daily windows from day 1 to day 7 would necessitate training at least seven distinct conversion models. This does not even account for scenarios beyond 7 days or finer-grained intervals such as hourly windows. Such a solution would lead to significantly increased training costs, high maintenance overhead, and greater risks to system stability.

Additional challenges arise in the production environment, including: 1) increased complexity in serving logic, as each optimization window would require distinct configurations to route requests to the correct models; 2) the need to build and maintain multiple data pipelines for label generation across different window lengths, which would impose high demands on both storage and computations.

Given these challenges, it is crucial to develop resource-efficient and scalable solutions that can flexibly support a wide range of optimization windows without incurring substantial system overhead.

4 Methodology

In this section, we first introduce the framework of the proposed *Personalized Interpolation* method and provide the underlying rationale behind its design. Next, in Section 4.2, we present several implementation variants of the interpolation mechanism and analyze their respective trade-offs. Finally, we describe how to seamlessly integrate the proposed method with existing black-box conversion prediction models in production environments.

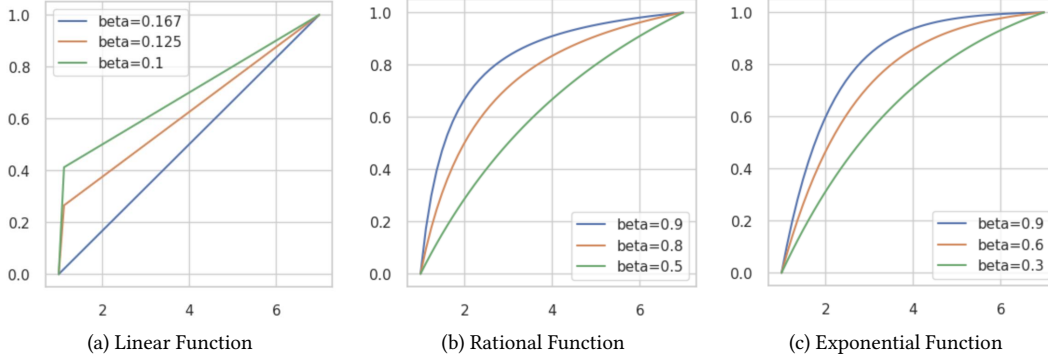


Figure 3: Illustration and comparison of the three proposed interpolation functions under the setting $T_s = 1$ and $T_l = 7$.

4.1 Framework of Personalized Interpolation

In our approach, we assume the availability of conversion probability predictions for two fixed optimization windows: $\mathbb{P}(\tau \leq T_s | e)$ and $\mathbb{P}(\tau \leq T_l | e)$, where T_s and T_l represent short and long optimization windows, respectively. Our goal is to estimate the conversion probability for an intermediate window $T_f \in (T_s, T_l)$.

Treating the conversion time τ as a random variable, the probability of conversion within any given window T corresponds to the cumulative distribution function (CDF), i.e., $\mathbb{P}(\tau \leq T | e)$. Since the CDF is a monotonically increasing function (see Figure 2), the conversion probability at window T_f can be approximated using a weighted combination of the two known predictions as follows:

$$\mathbb{P}(\tau \leq T_f | e) = (1 - \alpha(T_s, T_f, T_l)) \cdot \mathbb{P}(\tau \leq T_s | e) + \alpha(T_s, T_f, T_l) \cdot \mathbb{P}(\tau \leq T_l | e). \quad (4)$$

In Eq. (3), the interpolation coefficient $\alpha(T_s, T_f, T_l)$ corresponds to the normalized mass of the CDF between T_s and T_f :

$$\alpha(T_s, T_f, T_l) = \frac{\mathbb{P}(\tau \in (T_s, T_f] | e)}{\mathbb{P}(\tau \in (T_s, T_l] | e)} \in [0, 1].$$

However, the exact value of $\alpha(T_s, T_f, T_l)$ is typically unknown, as it depends on the underlying distribution of τ . Therefore, we treat α as a hyperparameter and interpret Eq. (3) as a general interpolation scheme that estimates $\mathbb{P}(\tau \leq T_f | e)$ using only the predictions at T_s and T_l . Since this interpolation is performed at the individual event level, we refer to this approach as *Personalized Interpolation*.

Furthermore, when the conditional probability $\mathbb{P}(\tau \in (T_s, T_l] | \tau > T_s)$ is of interest, the following proposition shows that, conditioning on no conversion occurring within the initial window T_s , the probability of conversion within the target window T_f is simply a scaled version of the probability within T_l :

PROPOSITION 1. *Assume predictions are available for two optimization windows T_s and T_l , and we seek to estimate the conversion probability for a new window $T_f \in [T_s, T_l]$. Then, the interpolation framework in Eq. (3) implies:*

$$\mathbb{P}(\tau \in (T_s, T_f] | \tau > T_s) = \alpha(T_s, T_f, T_l) \cdot \mathbb{P}(\tau \in (T_s, T_l] | \tau > T_s), \quad (4)$$

where $\mathbb{P}(\tau \in (T_s, T_f] | \tau > T_s)$ denotes the conditional probability of a conversion occurring in the interval $(T_s, T_f]$ given that no conversion occurred within T_s , and similarly for $\mathbb{P}(\tau \in (T_s, T_l] | \tau > T_s)$.

Proposition 1 suggests we can use a single scalar to interpolate conversions from one optimization window to another. This result

reinforces the central role of the interpolation factor $\alpha(T_s, T_f, T_l)$ in our personalized interpolation framework. For notational simplicity, we will henceforth refer to $\alpha(T_s, T_f, T_l)$ simply as α .

4.2 Design of Interpolations Factor

With the framework in Eq. (3), we now proceed to the design of the unknown interpolation factor $\alpha(T_s, T_f, T_l)$. In our work, we propose three types of parameterized functions: linear, rational, and exponential. Below, we describe each design in detail and compare them in Figure 2 using the setting $T_s = 1$ and $T_l = 7$:

- 1). *Linear function interpolation:* The simplest design adopts a linear form, where the interpolation factor is given by $\alpha(T_s, T_f, T_l) = \beta \cdot (T_f - T_l) + 1$, and β controls the slope (Fig. 3(a)). According to Eq. (4), this implies that $\mathbb{P}(\tau \in (T_s, T_f] | \tau > T_s)$ grows linearly toward $\mathbb{P}(\tau \in (T_s, T_l] | \tau > T_s)$. This design is suitable when the CDF exhibits a relatively flat curvature. To ensure $\alpha \in [0, 1]$, the feasible range for β is $[0, 1/(T_l - T_s)]$. However, when $\beta \neq 1/(T_l - T_s)$, the condition $\alpha(T_s, T_s, T_l) = 0$ introduces an abrupt drop near the boundary point T_s .
- 2). *Rational function interpolation:* This design introduces curvature into the interpolation using a rational function: $\alpha(T_s, T_f, T_l) = \frac{T_f - T_s}{\beta \cdot (T_f - T_l) + T_l - T_s}$, where β is a hyperparameter controlling the concavity (Fig. 3(b)). Unlike the linear case, this formulation ensures a smooth transition without abrupt changes at the boundary, making it suitable for more curved CDFs.
- 3). *Exponential function interpolation:* Assuming the conversion time τ follows a zero-inflated exponential distribution $\mathbb{P}(\tau \leq T_f | e) = p_{\text{conv}}(e) \cdot (1 - \exp(-\beta \cdot T_f))$, where $p_{\text{conv}}(e) = \mathbb{P}(\tau \leq \infty | e)$ is the ultimate conversion probability and $\beta \geq 0$, we can derive the interpolation factor as:

$$\alpha(T_s, T_f, T_l) = \frac{\exp(-\beta \cdot T_s) - \exp(-\beta \cdot T_f)}{\exp(-\beta \cdot T_s) - \exp(-\beta \cdot T_l)}.$$

As illustrated in Fig. 3(c), this design yields a smooth, concave interpolation curve, aligning well with scenarios where conversion likelihood decays exponentially over time.

In the above three designs of the interpolation factor, we introduce a hyperparameter β to control the shape of each interpolation function. Note that while we use the same symbol β across all designs, its meaning and effect differ depending on the specific formulation. In Section 5.2.3, we will conduct an ablation study to investigate the impact of β on estimation performance.

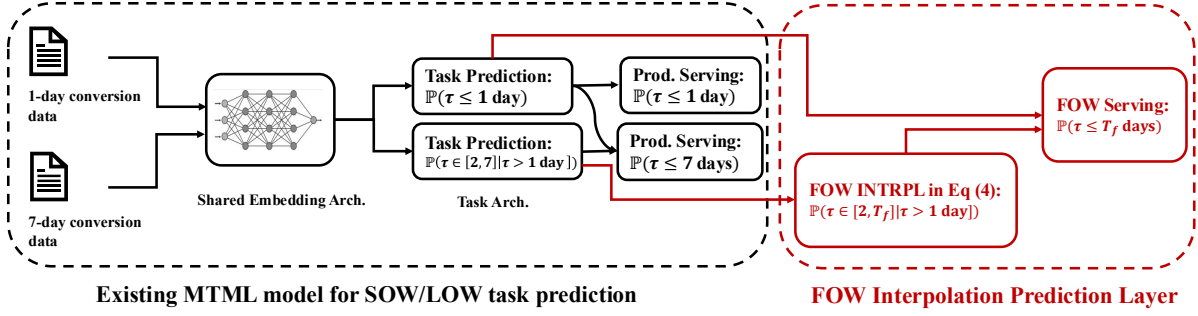


Figure 4: Model and system architecture of the SOW/LOW multi-task multi-label framework integrated with the flexible optimization window (FOW) interpolation layer.

4.3 Integration with Black-Box Models

We now describe how the proposed method can be seamlessly integrated with black-box conversion prediction models in production environments. In our interpolation framework (Eq. (3)), the conversion probabilities for the two optimization windows $\mathbb{P}(\tau \leq T_s|e)$ and $\mathbb{P}(\tau \leq T_l|e)$ are required. In practice, these can be obtained using either two separate deep neural network models or a single multi-task model, resulting in predictions denoted by $\widehat{\mathbb{P}}(\tau \leq T_s|e) = f_{T_s}(e)$ and $\widehat{\mathbb{P}}(\tau \leq T_l|e) = f_{T_l}(e)$.

Once the models are trained, an interpolation layer can be added on top of the outputs at inference time to compute the desired prediction for a flexible window T_f :

$$\widehat{\mathbb{P}}(\tau \leq T_f|e) = (1 - \alpha) \cdot f_{T_s}(e) + \alpha \cdot f_{T_l}(e),$$

where α is chosen according to a predefined interpolation design. This interpolation step is applied only during the evaluation stage and incurs no additional training or modeling cost.

Similarly, our approach can be applied in the conditional probability setting using Proposition 1. Assume the models can provide estimates of $\widehat{\mathbb{P}}(\tau \leq T_s|e) = f_{T_s}(e)$ and $\widehat{\mathbb{P}}(\tau \in (T_s, T_l] | \tau > T_s) = f_{T_l|T_s}(e)$. Then, by applying the law of total probability, we compute the flexible-window prediction as:

$$\widehat{\mathbb{P}}(\tau \leq T_f|e) = f_{T_s}(e) + (1 - f_{T_s}(e)) \cdot (\alpha \cdot f_{T_l|T_s}(e)).$$

We adopt this conditional probability formulation in our experiments and provide implementation details in Section 5.

5 Experiment

5.1 Experiment Setup

We conducted experiments using an ads conversion model based on a multi-task multi-label (MTML) consolidation framework [11, 19, 31], designed to predict conversions under two optimization windows: a short optimization window (SOW) of 1 day and a long optimization window (LOW) of 7 days. The overall model structure is illustrated in Fig. 4. The model takes merged conversion data from both the 1-day (SOW) and 7-day (LOW) windows as input, and outputs two predictions:

- The 1-day conversion probability, $\widehat{\mathbb{P}}(\tau \leq 1 \text{ day})$;
- The conditional probability of conversion in days 2 to 7, given no conversion in the first day, $\widehat{\mathbb{P}}(\tau \in [2, 7] | \tau > 1 \text{ day})$.

Using these two outputs, the overall 7-day conversion probability can be recovered by: $\widehat{\mathbb{P}}(\tau \leq 7 \text{ days}) = \widehat{\mathbb{P}}(\tau \leq 1 \text{ day}) + (1 - \widehat{\mathbb{P}}(\tau \leq 1 \text{ day})) \cdot \widehat{\mathbb{P}}(\tau \in [2, 7] | \tau > 1 \text{ day})$.

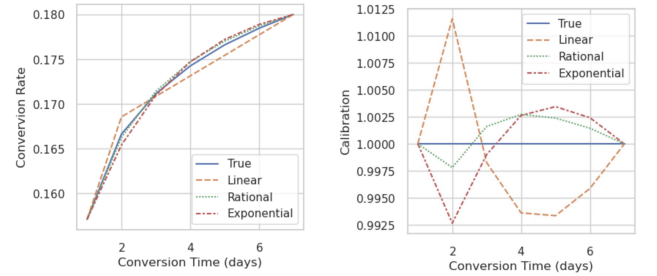


Figure 5: Conversion data analysis on real impression data collected from 2023-12-27 to 2024-02-04. The left panel shows the average conversion rates with three different interpolation estimations, while the right panel presents the corresponding calibration scores. For all three interpolation methods, the hyper-parameter β values are set as listed in Table 1.

Table 1: Hyper-parameter β values for interpolation methods in the experiments discussed in Sec. 5.2.1.

Type	Linear	Rational	Exponential
β value	0.1	0.7	0.4

1 day) $\cdot \widehat{\mathbb{P}}(\tau \in [2, 7] | \tau > 1 \text{ day})$. This MTML consolidation approach provides an efficient solution to the long optimization window problem. In our experiments, we use the existing MTML model to estimate conversion probabilities for flexible optimization windows (FOW) with $T_f \in (1, 7)$ using the proposed interpolation framework.

In our experiments, the training data is the impression data collected from ads delivery system over 40 days. Before model training and evaluation, we performed an analysis of the conversion time distribution. The left panel of Fig. 5 depicts the true conversion probability curve (i.e., the CDF of τ) as the solid blue line. The curve exhibits a concave shape, increasing from $\mathbb{P}(\tau \leq 1 \text{ day}) = 0.157$ to $\mathbb{P}(\tau \leq 7 \text{ days}) = 0.180$. Using these true values at the 1-day and 7-day windows, we apply the proposed interpolation methods and plot the estimated conversion probabilities as dashed curves. These interpolated curves closely follow the true conversion curve. To quantitatively assess the difference between the true and interpolated conversion rates, we show calibration scores in the right panel of Fig. 5. The calibration score [10] is defined as the ratio between the interpolated estimation, $\widehat{\mathbb{P}}_{\text{INTRPL}}(\tau \leq T_f \text{ day})$, and the true value, $\mathbb{P}(\tau \leq T_f \text{ day})$, for $T_f \in [1, 7]$. The results indicate that

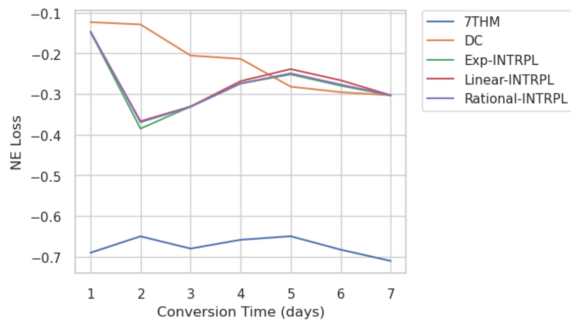


Figure 6: Normalized entropy (NE) loss (%) compared with NDUB. The less negative value means better performance.

all three interpolation methods yield calibration scores close to 1, with the rational function interpolation exhibiting the smallest deviation. Based on this conversion data analysis, we are confident that our proposed interpolation methods provide an accurate and reliable fit for flexible optimization window (FOW) prediction.

5.2 Performance Comparison

5.2.1 Comparison with Existing methods. Now, we proceed to evaluate our proposed interpolation methods by comparing them against five competitive baselines:

- \mathbb{P}_{1d} -Model: This method directly uses the SOW prediction $\mathbb{P}(\tau \leq 1 \text{ day})$ as an estimate for $\mathbb{P}(\tau \leq T_f \text{ day})$ for all $T_f \in [1, 7]$. Since it ignores potential conversions beyond 1 day, it tends to be under-calibrated (i.e., calibration scores below 1).
- \mathbb{P}_{7d} -Model: This method treats the LOW prediction $\mathbb{P}(\tau \leq 7 \text{ day})$ as a proxy for $\mathbb{P}(\tau \leq T_f \text{ day})$ for $T_f \in [1, 7]$. It overestimates early conversions, resulting in over-calibration (i.e., calibration scores above 1).
- 7-TaskHead Model (7THM): A newly designed multi-task model with 7 separate heads, each trained to predict conversion by day T_f for $T_f = 1$ to 7. The model’s effectiveness depends on proper task balancing and sufficient training data for each head.
- Dedicated Models (DMs): This approach trains 6 individual multi-task models. Each model predicts both $\mathbb{P}(\tau \leq 1 \text{ day})$ and the conditional probability $\mathbb{P}(\tau \in T_f \text{ to } 7 \text{ days} \mid \tau > 1 \text{ day})$ for a specific $T_f \in [2, 7]$, which are composite into $\mathbb{P}(\tau \leq 7 \text{ day})$. However, the training labels for these models are delayed, reflecting real production constraints.
- No Delay Upper Bound (NDUB): Identical to DMs in structure, but trained with up-to-date (non-delayed) conversion labels. This method is not feasible in production but serves as a theoretical upper bound on achievable performance.

For our interpolation-based methods, we use the hyper-parameter values in Table 1, selected through simple tuning. An ablation study analyzing the impact of β is provided in Section 5.2.3.

Experimental Results. We evaluate prediction accuracy using the normalized entropy (NE) metric [10], where lower NE values indicate better performance. The NE results for all methods are reported in Table 2. Our three interpolation methods achieve NE scores that closely match those of the Dedicated Models (DMs) baseline, indicating that the proposed interpolation framework is

highly effective for production deployment. Unlike DMs, which requires training six separate models, our approach only requires a single light-weight model, substantially reducing both training and infrastructure costs. The method using \mathbb{P}_{1d} model yields the worst NE performance for estimating conversions beyond the first day, due to its under-calibrated nature. Conversely, the \mathbb{P}_{7d} model approximation performs poorly in the early part of the conversion window (first three days) but improves as the prediction horizon lengthens. These observations align with the trend in Fig. 5, where the empirical conversion CDF is concave and flattens over time. The 7-TaskHead Model (7THM) produces moderately competitive NE values, though consistently worse than those of our interpolation framework. We attribute this to the difficulty of maintaining balance across multiple tasks, which can hinder the performance improvements of such multi-head architectures.

To further analyze the performance gap between our proposed method and the theoretical upper bound, we use the results of the No Delay Upper Bound (NDUB) method as a baseline and compute the NE loss (%) for the Dedicated Models (DMs) method, the 7-TaskHead Model (7THM), and our three interpolation methods. The results are visualized in Figure 6. The DC method consistently delivers the best performance, with NE loss increasing slightly from -0.1% to -0.3% as the conversion window extends from 1 day to 7 days. Our interpolation methods show an NE improvement of approximately -0.3% for conversion windows beyond 1 day. Interestingly, as the prediction horizon increases, our methods begin to outperform the DMs approach, highlighting their strengths in both prediction accuracy and training efficiency. It is important to note that this training efficiency arises from the fact that the DMs method requires training six separate models, whereas our interpolation framework only requires a single model. In contrast, the 7THM approach exhibits significantly larger NE losses, around -0.7% , indicating suboptimal performance. Overall, these comparative results demonstrate that our proposed interpolation framework achieves a favorable trade-off between predictive accuracy and computational efficiency, making it particularly well-suited for large-scale deployment.

5.2.2 Performance Check with Recurring Training. To evaluate the long-term effectiveness of our methods, we conducted recurring training and evaluation over a one-week period, spanning from 2024-02-11 to 2024-02-17. Each day’s model was trained using the most recent data and evaluated on the following week. In this analysis, we use the 7-TaskHead model as a baseline and compare it against our personalized interpolation methods. Fig. 7 presents the normalized entropy (NE) gains (%) achieved by our methods relative to the baseline. The results show that our interpolation methods consistently outperform the 7-TaskHead model, achieving over 0.2% NE gain across the evaluation period. Notably, the NE differences among the three interpolation variants are minimal, suggesting all designs offer stable and robust performance. Additionally, all three interpolation methods yield identical NE values at the 1-day and 7-day boundaries, since their interpolated estimates align exactly with the original predictions at these endpoints.

5.2.3 Ablation Study on Hyperparameters. In the final part of our experiments, we conducted an ablation study to investigate the influence of the hyper-parameter β on the accuracy and stability of

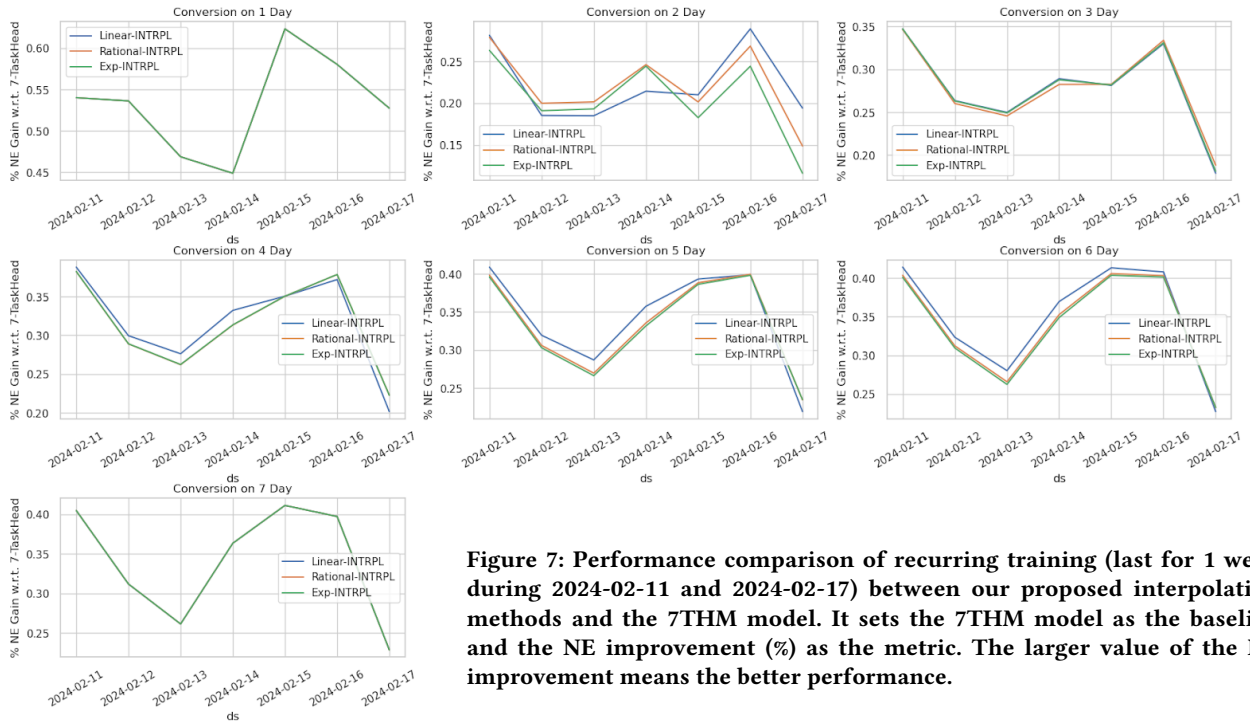


Figure 7: Performance comparison of recurring training (last for 1 week during 2024-02-11 and 2024-02-17) between our proposed interpolation methods and the 7THM model. It sets the 7THM model as the baseline and the NE improvement (%) as the metric. The larger value of the NE improvement means the better performance.

Table 2: Performance comparison of different methods for flexible optimization window (FOW) prediction. The table reports the normalized entropy (NE) values across various conversion windows. Lower NE values indicate better predictive performance.

Conv. Day	Methods							
	NDUB	DMs	Linear-INTRPL	Rational-INTRPL	Exp-INTRPL	7THM	P_{1d}	P_{7d}
1	0.5995	0.6002	0.6004	0.6004	0.6004	0.6036	0.6004	0.6159
2	0.6051	0.6059	0.6073	0.6073	0.6074	0.6090	0.6177	0.6115
3	0.6082	0.6095	0.6103	0.6103	0.6103	0.6124	0.6238	0.6123
4	0.6110	0.6123	0.6126	0.6127	0.6127	0.6150	0.6287	0.6137
5	0.6130	0.6148	0.6145	0.6146	0.6146	0.6170	0.6326	0.6150
6	0.6144	0.6162	0.6161	0.6161	0.6161	0.6186	0.6360	0.6163
7	0.6156	0.6175	0.6175	0.6175	0.6175	0.6200	0.6391	0.6175

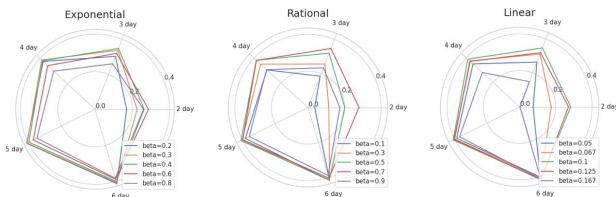


Figure 8: Ablation study on the hyper-parameter β in our interpolation methods. The 7-TaskHead model serves as the baseline, and the NE improvement (%) is reported as the evaluation metric. A larger shaded area indicates better overall accuracy and robustness to hyper-parameter variation.

our interpolation methods. Since β is the only tunable parameter in our framework, we systematically varied its value and measured the corresponding NE gain (%) over the 7-TaskHead baseline. The results, shown in Figure 8, reveal that the exponential interpolation

method is relatively less sensitive to changes in β , demonstrating greater robustness compared to the linear and rational function interpolations. Furthermore, for all interpolation types, the NE improvement diminishes as the prediction window approaches 1 day, which aligns with the reduced role of interpolation near the lower boundary.

6 Conclusion

In this paper, we proposed a novel framework, termed *personalized interpolation*, to effectively address the Flexible Optimization Window (FOW) problem in advertising conversion modeling. The framework leverages two conversion predictions corresponding to short and long optimization windows and interpolates between them to estimate conversion probabilities for arbitrary windows, thereby removing the need for direct supervision. Our method is model-agnostic and can be seamlessly integrated with existing black-box

conversion prediction systems commonly used in industrial environments, with negligible extra computational or training overhead. Extensive experiments demonstrate that our interpolation-based approach consistently achieves competitive or superior predictive performance compared to alternative production-ready solutions. These results validate the effectiveness and practicality of the proposed method for real-world deployment in advertising systems.

References

- [1] Nicola Barbieri, Fabrizio Silvestri, and Mounia Lalmas. 2016. Improving post-click user engagement on native ads via survival analysis. In *Proceedings of the 25th International Conference on World Wide Web*. 761–770.
- [2] Norman E Breslow. 1975. Analysis of survival data under the proportional hazards model. *International Statistical Review/Revue Internationale de Statistique* (1975), 45–57.
- [3] Olivier Chapelle. 2014. Modeling delayed feedback in display advertising. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 1097–1105.
- [4] Yu Chen, Jiaqi Jin, Hui Zhao, Pengjie Wang, Guojun Liu, Jian Xu, and Bo Zheng. 2022. Asymptotically unbiased estimation for delayed feedback modeling via label correction. In *Proceedings of the ACM Web Conference 2022*. 369–379.
- [5] David R Cox. 1972. Regression models and life-tables. *Journal of the Royal Statistical Society: Series B (Methodological)* 34, 2 (1972), 187–202.
- [6] Carl De Boor and Carl De Boor. 1978. *A practical guide to splines*. Vol. 27. Springer New York.
- [7] Regina C Elandt-Johnson and Norman L Johnson. 2014. *Survival models and data analysis*. John Wiley & Sons.
- [8] Walter Gautschi. 2011. *Numerical analysis*. Springer Science & Business Media.
- [9] Robert J Gray. 1992. Flexible methods for analyzing survival data using splines, with applications to breast cancer prognosis. *J. Amer. Statist. Assoc.* 87, 420 (1992), 942–951.
- [10] Xinran He, Junfeng Pan, Ou Jin, Tianbing Xu, Bo Liu, Tao Xu, Yanxin Shi, Antoine Atallah, Ralf Herbrich, Stuart Bowers, et al. 2014. Practical Lessons from Predicting Clicks on Ads at Facebook. In *Proceedings of the Eighth International Workshop on Data Mining for Online Advertising*. ACM.
- [11] Yan Huang, Wei Wang, Liang Wang, and Tieniu Tan. 2013. Multi-task deep neural network for multi-label learning. In *2013 IEEE International conference on image processing*. IEEE, 2897–2900.
- [12] Stephen P Jenkins. 2005. Survival analysis. *Unpublished manuscript, Institute for Social and Economic Research, University of Essex, Colchester, UK* 42 (2005), 54–56.
- [13] Wendi Ji, Xiaoling Wang, and Dell Zhang. 2016. A probabilistic multi-touch attribution model for online advertising. In *Proceedings of the 25th ACM international conference on information and knowledge management*. 1373–1382.
- [14] Edward L Kaplan and Paul Meier. 1958. Nonparametric estimation from incomplete observations. *Journal of the American statistical association* 53, 282 (1958), 457–481.
- [15] Jared L Katzman, Uri Shaham, Alexander Cloninger, Jonathan Bates, Tingting Jiang, and Yuval Kluger. 2018. DeepSurv: personalized treatment recommender system using a Cox proportional hazards deep neural network. *BMC medical research methodology* 18 (2018), 1–12.
- [16] Sofia Ira Ktena, Alykhan Tejani, Lucas Theis, Pranay Kumar Myana, Deepak Dilipkumar, Ferenc Huszár, Steven Yoo, and Wenzhe Shi. 2019. Addressing delayed feedback for continuous training with neural networks in CTR prediction. In *Proceedings of the 13th ACM conference on recommender systems*. 187–195.
- [17] Alexandros Labrinidis and Nick Roussopoulos. 2003. Balancing performance and data freshness in web database servers. In *Proceedings 2003 VLDB Conference*. Elsevier, 393–404.
- [18] Kuang-Chih Lee, Ali Jalali, and Ali Dasdan. 2013. Real time bid optimization with smooth budget delivery in online advertising. In *Proceedings of the seventh international workshop on data mining for online advertising*. 1–9.
- [19] Ning Ma, Mustafa Ispir, Yuan Li, Yongpeng Yang, Zhe Chen, Derek Zhiyuan Cheng, Lan Nie, and Kishor Barman. 2022. An online multi-task learning framework for google feed ads auction models. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 3477–3485.
- [20] Nicolas Molinari, Jean-Pierre Daurès, and Jean-François Durand. 2001. Regression splines for threshold selection in survival data analysis. *Statistics in medicine* 20, 2 (2001), 237–247.
- [21] Milad Zafar Nezhad, Najibesadat Sadati, Kai Yang, and Dongxiao Zhu. 2019. A deep active survival analysis approach for precision treatment recommendations: Application of prostate cancer. *Expert Systems with Applications* 115 (2019), 16–26.
- [22] Ping Wang, Yan Li, and Chandan K Reddy. 2019. Machine learning for survival analysis: A survey. *ACM Computing Surveys (CSUR)* 51, 6 (2019), 1–36.
- [23] Yifan Wang, Peijie Sun, Min Zhang, Qinglin Jia, Jingjie Li, and Shaoping Ma. 2023. Unbiased Delayed Feedback Label Correction for Conversion Rate Prediction. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2456–2466.
- [24] Yanshi Wang, Jie Zhang, Qing Da, and Anxiang Zeng. 2020. Delayed feedback modeling for the entire space conversion rate prediction. *arXiv preprint arXiv:2011.11826* (2020).
- [25] Lee-Jen Wei. 1992. The accelerated failure time model: a useful alternative to the Cox regression model in survival analysis. *Statistics in medicine* 11, 14-15 (1992), 1871–1879.
- [26] Alice S Whittemore and Joseph B Keller. 1986. Survival estimation using splines. *Biometrics* (1986), 495–506.
- [27] Jia-Qi Yang, Xiang Li, Shuguang Han, Tao Zhuang, De-Chuan Zhan, Xiaoyi Zeng, and Bin Tong. 2021. Capturing delayed feedback in conversion rate prediction via elapsed-time sampling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 4582–4589.
- [28] Shota Yasui and Masahiro Kato. 2022. Learning Classifiers under Delayed Feedback with a Time Window Assumption. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 2286–2295.
- [29] Shota Yasui, Gota Morishita, Fujita Komei, and Masashi Shibata. 2020. A feedback shift correction in predicting conversion rates under delayed feedback. In *Proceedings of The Web Conference 2020*. 2740–2746.
- [30] Yuya Yoshikawa and Yusaku Imai. 2018. A nonparametric delayed feedback model for conversion rate prediction. *arXiv preprint arXiv:1802.00255* (2018).
- [31] Yu Zhang and Qiang Yang. 2018. An overview of multi-task learning. *National Science Review* 5, 1 (2018), 30–43.

Received 16 May 2025; revised 16 May 2025; accepted 16 May 2025