DART - Drift Aware Ranking with forecasTing and uncertainty

Deep Nayak
Amazon
Bengaluru, India
deepnyk@amazon.com

Shreyas S
Amazon
Bengaluru, India
shreyas.sheshadhri@gmail.com

Sivaramakrishnan Kaveri
Amazon
Bengaluru, India
kavers@amazon.com

Abstract

Ranking systems in online platforms across domains must handle temporal drift where artifact distributions and contextual relevance evolve over time. Existing approaches address temporal drift using recency weighting, sliding windows, or forecasting techniques, however, they do not focus on integrating drift signals with ranking systems or handle the data variability introduced by the drift. We propose DART, a two-stage architecture combining a ranking model with a forecasting model that predicts artifact performance for a look-ahead window. An uncertainty-aware attention mechanism dynamically weights forecasted information based on prediction confidence, enabling robust adaptation without online updates to the ranker at serving time. Prior work uses uncertainty either to detect drift [19] or to debias ranking [11]; DART instead uses it as a control signal that modulates information flow from a forecaster to the ranker. Experiments on a large-scale e-commerce promotion ranking dataset show DART's deployable offline configuration achieves +11.11% ± 0.11% relative Mean Reciprocal Rank (MRR) during promotional events and +7.37% ± 0.11% pre-event. On the public Coveo SIGIR dataset DART delivers ∼+22% MRR in both high and low-drift periods, showing dual-regime robustness across abrupt and gradual drift; the mechanism is task-agnostic and also improves a regression benchmark. Ablations demonstrate that attention-based integration of forecasted information substantially outperforms direct feature inclusion, reinforcing the importance of uncertainty-aware temporal adaptation in ranking systems.

CCS Concepts

• **Computing methodologies**; • **Information systems** → **Learning to rank**;

Keywords

Ranking, Forecasting, Domain Drift, Uncertainty

ACM Reference Format:
Deep Nayak, Shreyas S, and Sivaramakrishnan Kaveri. 2026. DART - Drift Aware Ranking with forecasTing and uncertainty. In *20th ACM Conference on Recommender Systems (RecSys '26), September 27-October 02, 2026, Minneapolis, MN, USA*. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3773078.3831908

RecSys '26, Minneapolis, MN, USA

ACM ISBN 979-8-4007-2284-4/2026/09
https://doi.org/10.1145/3773078.3831908

1 Introduction

Ranking systems must rank artifacts within specific contexts while handling temporal drift, where artifact distributions and contextual relevance evolve over time. Consider vacation rental platforms ranking properties based on traveller preferences and seasonal demand, where beachfront and mountain properties show clear seasonal patterns. Content platforms rank articles considering user interests and trending topics, with categories tied to news cycles. E-commerce stores exemplify this challenge through promotional offer ranking. When customers browse products or proceed to checkout, stores must decide which promotional incentives to display prominently. Should a customer see a **Financial Partner Discount** from their preferred bank, or would a **Bundle Savings** offer encouraging additional purchases be more compelling? These decisions directly impact whether customers complete their purchases and how much they spend.

The challenge intensifies during major temporal shifts across domains. During festival sales, e-commerce stores see financial partner discounts dominate as customers make high-value purchases, with some partners aggressively adding higher discount offers while others completely remove offerings (figure 1). Content platforms experience dramatic shifts during breaking news events or catastrophes, where certain article categories suddenly surge in relevance while others decline. Vacation rental platforms witness property distribution changes during peak travel seasons, with beach properties becoming scarce in summer while mountain listings increase in winter. Each temporal period requires different ranking strategies to maximize both user satisfaction and business objectives.

Traditional static ranking models struggle in these dynamic settings as they cannot adapt to evolving distributions. Current approaches to temporal drift span reactive and predictive techniques. Recency-based methods such as sliding windows and exponential decay weighting [12] down-weight older observations but cannot anticipate future shifts. Forecasting-based methods predict future artifact performance: offline methods use prediction variance as a drift indicator [19, 33] while online methods [2, 31] adapt through ensemble techniques and gradient updates. However, significant challenges remain: (1) recency methods are purely reactive and miss anticipatory behavior (e.g., customers delaying purchases before sales); (2) forecasting approaches optimize prediction accuracy rather than downstream ranking, and online methods require continuous updates unsuitable for production systems under strict latency constraints; (3) none account for heteroscedastic uncertainty, i.e., the varying reliability of predictions across temporal regimes.

To address the above challenges, we investigate the following research questions: **RQ1:** How can we integrate drift information captured by a forecasting model with a light-weight ranking model?

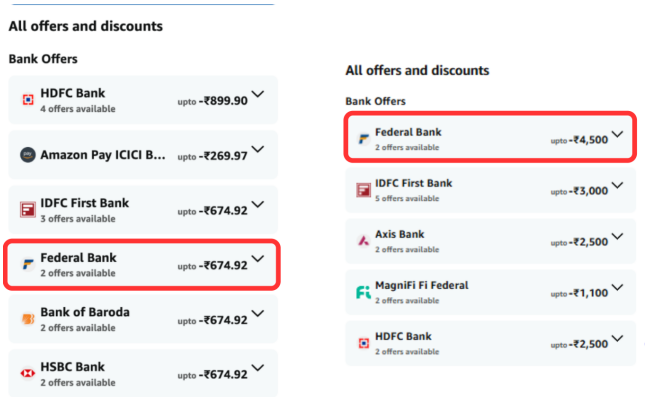

Figure 1: Left: promotional offers by several financial partners shown to a customer for an electronics product in Jan 2025. Right: promotional offers for the same product/customer during a major shopping event (Oct 2025). Red box highlights the offer which was redeemed by the customer. Note that "personalization" in ranking is necessary as simple heuristics like sorting by discount will not be ideal.

We perform multiple ablations quantifying the impact of using forecasted signals and representations as raw input in ranking models. Our experiments show that using an attention layer to integrate forecasted representations with the ranking models yield the best results. **RQ2:** How do we handle data variability introduced by drift across time periods when integrating forecasted information? We focus our experiments to verify if this data variability can be addressed by using epistemic uncertainty measures and highlight the effectiveness of using uncertainty aware attention to integrate information between two models.

Contributions: In this paper, we focus our experiments on the primary task of ranking promotional offers in the e-commerce domain. Our contributions can be summarized as follows: (1) We formalize temporal drift challenges in ranking systems by characterizing how promotion distributions change over time in e-commerce environments; (2) We propose **DART**, which integrates drift-aware forecasting models with ranking models with no online ranker updates at serving time; the forecaster supports test-time-adaptive variants but is deployed as a periodic batch refresh in production (RQ1); (3) We introduce an uncertainty-aware attention mechanism that dynamically adjusts how much the ranking model trusts forecasted information based on prediction confidence, ensuring robust performance when data quality varies (RQ2); (4) We design DART for production deployment under stringent latency constraints (<10ms inference budget) at scale, using a two-stage architecture where the forecaster runs in batch with cached embeddings, adding minimal overhead (one attention layer plus a small uncertainty-scaling MLP) on the real-time path. DART's contribution lies in the uncertainty-scaled attention mechanism that governs information flow between a forecaster and a ranker: rather than naively concatenating forecasted signals or relying on a simple two-stage pipeline, DART uses epistemic uncertainty to dynamically modulate how much the ranker trusts temporal predictions through learned attention.

Experiments on a large-scale e-commerce promotion ranking system show DART's deployable offline configuration delivers **+11.11%** ± **0.11%** relative MRR during events and **+7.37%** ± **0.11%** pre-event versus static baselines (online OneNet upper bound: +11.99% / +7.94%). On Coveo SIGIR [26] DART delivers ∼+22% MRR in both high- and low-drift periods (dual-regime robustness without retuning). These improvements translate to higher conversion rates and more relevant promotions. While our primary focus addresses e-commerce promotion ranking, DART's approach to temporal drift applies broadly to any downstream task in any domain.

2 Related Work

Our work intersects attention mechanism for integrating complementary information, uncertainty estimation in ranking systems, and temporal drift handling in forecasting. Each addresses different aspects of temporal adaptation: attention for integration, uncertainty for reliability estimates, and forecasting for temporal patterns.

Attention for complementary information integration: Existing work uses attention to incorporate external information through knowledge distillation where teacher models guide students via attention alignment [13, 15, 17], multi-passage fusion where retrieved documents are combined through cross-attention for question answering [6], structured knowledge injection where graph embeddings are integrated into vision-language models via adapter attention [16], and Bayesian context updates where energy-based modules provide posterior-updated contexts to diffusion models [23]. However, these approaches focus on static information transfer rather than temporal adaptation. DART extends this by using uncertainty-weighted attention to dynamically integrate forecasted temporal patterns based on prediction reliability.

Uncertainty in Ranking: Prior work applies uncertainty through probabilistic language models that use variance as risk components in information retrieval [33] and improve fairness in ranking models [7, 11, 28]. Uncertainty has also been used for drift detection using prediction confidence to identify temporal shifts [19] and model error variations to quantify distribution changes [4, 8]. These methods primarily either use uncertainty for reducing bias in ranking or detect drift rather than adapt ranking decisions to it. Our approach uses uncertainty estimation to dynamically weigh forecasted information based on confidence intervals and historical data availability patterns.

Temporal Drift in Ranking and Forecasting: Recency-based methods adapt through sliding windows and importance weighting with exponential decay [12], but are purely backward-looking. Online forecasting handles shifts through gradient-based updates [2], adaptive learning rates [18], synthetic data generation [30] and ensemble methods [31] with transformer architectures [22, 29, 32], but these optimize prediction accuracy rather than downstream ranking. In summary, while prior work addresses recency weighting for non-stationarity, attention for information fusion, uncertainty for reliability estimation, and forecasting for temporal patterns, no existing approach combines these elements for drift-aware ranking with explicit uncertainty calibration. DART's key distinction is

treating uncertainty as a first-class signal that governs information integration between a forecaster and a ranker.

3 Problem Setting and Temporal Drift

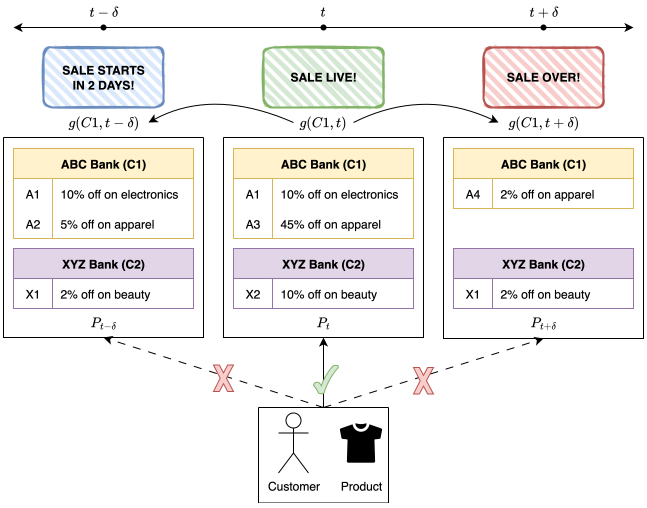

Figure 2: Example of artifact grouping and volatile nature of individual artifacts. Depicts the dependence on $t \pm \delta$ and drift in label since customers withhold purchases in anticipation of sale events.

Let $\mathcal{X}$ denote the universe of contexts, $\mathcal{P}$ the universe of artifacts. Let t denote a specific point in time over the horizon 0 to T. We define $\mathcal{P}_t$ as the set of artifacts that are available at time t and $\mathcal{X}_t$ as a sample of contexts at time t. For each artifact $p_t \in \mathcal{P}_t$, context $x_t \in \mathcal{X}_t$ at time $t \in \mathcal{T}$, let $y_{x_t,p_t} \in \{0, 1\}$ denote the binary feedback for the tuple (x_t, p_t). In the e-commerce domain, for the task of promotion ranking, $\mathcal{P}$ translates to all possible promotions introduced on the store since its inception, $\mathcal{P}_t$ would be the set of promotions visible to customers at time t, $\mathcal{X}_t$ represents the active customer and product set on the platform at time t and y_{x_t,p_t} denotes the feedback for each individual context, promotion pair. For the e-commerce promotion ranking task, y_{x_t,p_t} is positive when a customer "redeems" or uses a promotion while purchasing a product. Our objective is to generate a personalized ranked list $\pi_{x_t,\mathcal{P}_t}$ of all artifacts $p_t \in \mathcal{P}_t$ for a given context x_t at time t, derived by scoring each (x_t, p_t) pair independently via a pointwise model and sorting by score.

Problem of Temporal Drift: Due to the dynamic nature of artifact availability, we observe that the binary feedback y_{x_t,p_t} becomes incoherent even in the presence of perfect context and artifact information. This is because in the presence of temporal drift, y_{x_t,p_t} depends not only on $\mathcal{P}_t$ but also on the past and future states of the artifact set denoted by $\mathcal{P}_{t+\delta}$ and $\mathcal{P}_{t-\delta}$. It is quite possible that p_t does not even exist at $t \pm \delta$. To illustrate, consider a standard "5% off" promotion (p_t) on apparel. As shown in figure 2, in the weeks preceding major sales, this promotion experiences a drop in usage as customers postpone purchases in anticipation of deeper discounts, *despite no change to the promotion itself or its features* (discount percentage, financial partner, etc. remain identical). A pointwise ranker with access to all promotion attributes cannot capture this behavior because the drift originates from the temporal context, not the artifact features. During the sale, this promotion may be replaced with a "45% off" offer, further shifting the distribution.

4 Modeling temporal drift in e-commerce using DART

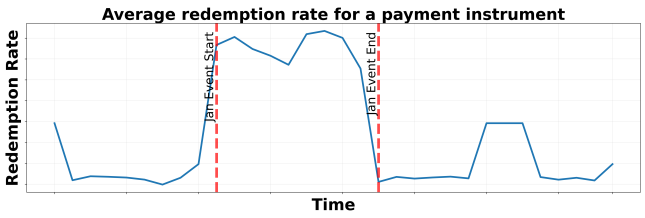

Figure 3: Average redemption rate of promotions of a financial partner, before, during, and after a major promotional event (denoted by red markers).

We divide our modeling discussion into smaller sections focusing on different components that work together to build DART.

Ranking model (ranker): We generate a personalized ranked list $\pi_{x_t,\mathcal{P}_t}$ using a point-wise ranker trained on the binary classification task with y_{x_t,p_t}, the redeem signal, as the target label. The ranking model generates a score for every promotion, context pair and we derive $\pi_{x_t,\mathcal{P}_t}$ by sorting p_t by their score in a descending order. We also define $f'_\theta(x_t, p_t)$ as the ranker's representation before final scoring output layer, its usage in DART's context is outlined in later sections. The vanilla point wise ranking model proves to be insufficient in ranking artifacts under temporal drift, hence we define a forecasting model to estimate the drift in the artifacts.

Forecasting model (forecaster): Quantifying drift at the individual artifact level is infeasible because artifacts are ephemeral and turn over on timescales shorter than the forecasting horizon, while the "grouping" they belong to (e.g. category, type) remains relatively consistent. Hence it becomes imperative to define a metric similar to y_{x_t,p_t} at the group level in order to measure the drift in artifacts across time. Let $c \in C$ denote a particular grouping and g denote an aggregation function of y_{x_t,p_t} over all p_t who belong to the group c at time t. In the e-commerce domain, we define c as the financial partner offering promotions and g as the redemption rate (RR). Hence, $g(c, t) = \frac{\sum_{p_t \in c} \sum_{x_t} \mathbb{1}[y_{x_t,p_t}=1]}{\sum_{p_t \in c} \sum_{x_t} \mathbb{1}[\text{shown}(x_t,p_t)]}$, where the numerator counts redemptions and the denominator counts impressions of promotions in group c at time t. As observed in figure 3, the drift of promotion set can be quantified using the variation in the redemption rate of a group of promotions offered by a particular financial partner. We define historical features that include historical redemption rates $g(c, t-\tau)$ for each financial partner where τ represents a rolling window ranging from 1 day to 1 month to capture historical patterns at different granularity. We also include historical customer cohort distribution and historical distribution of promotions across categories as additional input signals to our forecaster.

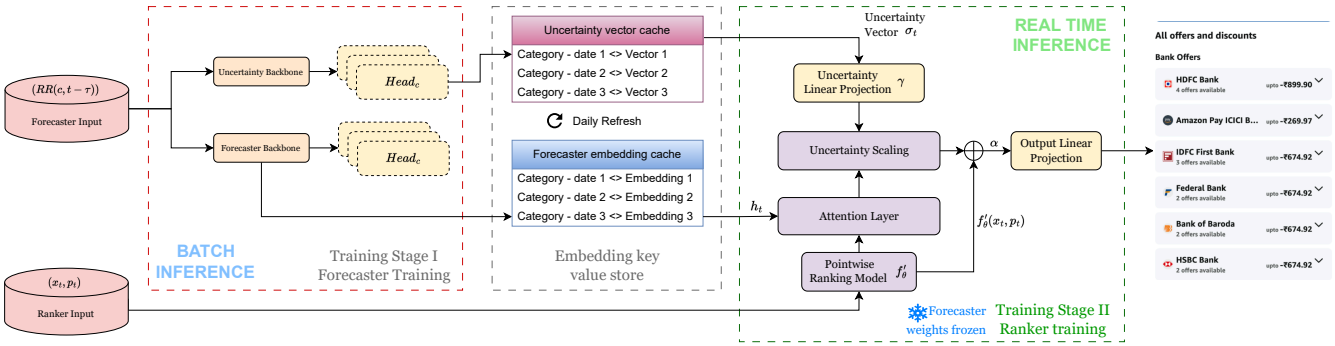

Figure 4: Stage I trains the forecaster and uncertainty model followed by Stage II where pointwise ranker is trained with uncertainty scaled attention. The left half denotes offline components which store output in a cache whereas the right half shows components serving requests in real time.

The forecaster uses these features to predict future redemption rates: $\widehat{g}(c, t+\delta) = \text{Head}_c(\mathbf{h}_t[c])$, where $\mathbf{h}_t \in \mathbb{R}^{|C| \times d}$ stacks per-group embeddings produced in parallel by a shared encoder, $\mathbf{h}_t[c] \in \mathbb{R}^d$ is the row for group c, δ is the look-ahead hyperparameter, and Head_c is a group-specific output layer. This multi-task learning approach results in two major advantages: (1) a common backbone and joint learning allows knowledge sharing between different forecaster heads improving their accuracy (2) we are able to extract a shared representation of promotions depending on the context which encapsulates drift information and allows easy integration with other models

Attention Mechanism: In order to improve the ranker's performance using drift information captured by the forecaster, we need to define an integration mechanism. A simple technique is to pass the forecaster's output as input to the ranker. However, our experiments (section 5) demonstrate that for the e-commerce promotion ranking task, directly passing the forecaster's output or its hidden representation $\mathbf{h}_t$ to the ranker proves to be ineffective. We propose to use an attention layer to integrate the forecasted information into the ranker. After training the forecaster, we discard the group specific output heads and extract shared latent $\mathbf{h}_t$ embedding. Since the forecaster maintains separate prediction heads for each group $c \in C$, the shared backbone produces $\mathbf{h}_t \in \mathbb{R}^{|C| \times d}$. Note that $\mathbf{h}_t$ is computed once per time step (shared across all queries at time t); per-query personalisation comes from the attention query $f'_\theta(x_t, p_t)$, which also enables the cached-embedding deployment in section 6. The ranker's penultimate representation $f'_\theta(x_t, p_t)$ serves as the query and attends over this sequence as keys and values via multi-head attention: $\text{Att}(f'_\theta(x_t, p_t), \mathbf{h}_t, \mathbf{h}_t)$, allowing the model to selectively weight drift information from different groups based on the current context-promotion pair.

Need for uncertainty estimation: While the forecaster embedding provides valuable future-state information to the ranker, its reliability varies across time. We observe that uncertainty peaks during pre-event and post-event windows, plausibly driven by both data sparsity (fewer interaction logs in the run-up to and after major events) and inherent regime unpredictability (transitional behaviour is harder to forecast regardless of sample size). This data sparsity directly impacts the quality of the forecaster embedding $\mathbf{h}_t$

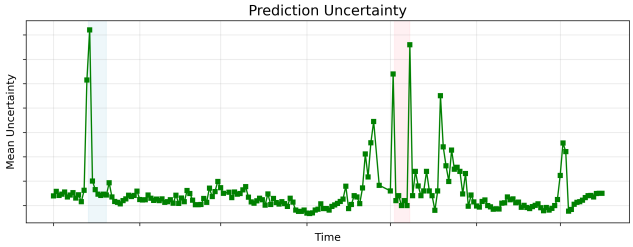

Figure 5: Prediction uncertainty across time. Shaded regions denote major promotional events and peaks are observed right before and after them.

and may introduce noise into the attention mechanism. To address this issue, we explicitly model the time-dependent uncertainty in our forecasting predictions. As illustrated in Figure 5, uncertainty exhibits clear temporal patterns aligned with promotional event cycles. By quantifying this uncertainty, we can inform the ranking model about the reliability of forecaster embedding it receives. Let σ_t represent the uncertainty estimate at time t. We integrate this uncertainty into our attention mechanism as follows:

$$s(x_t, p_t) = \alpha \cdot f'_\theta(x_t, p_t) + (1-\alpha) \cdot \text{Att}(f'_\theta(x_t, p_t), \mathbf{h}_t, \mathbf{h}_t) \cdot \gamma(\sigma_t)^{-1} \quad (1)$$

Where α is a hyperparameter that controls the balance between the original ranker output and the attention-modified output.

Role and design of $\gamma(\cdot)$: γ translates the raw uncertainty σ_t into a scaling factor for the attention output. It serves two roles: **(i) calibration:** the scale of σ_t is estimator-dependent (TabM std., MC-dropout variance, etc.) and unrelated a priori to the attention-output magnitude, so a fixed σ_t^{-1} would over or under-suppress the forecast; γ learns the right per-dimension calibration from the ranking loss. **(ii) selectivity:** since $\sigma_t \in \mathbb{R}^d$ is dimension-aligned with $\mathbf{h}_t$, a non-linear γ attenuates individual forecasted components rather than collapsing to a scalar gate. We use a small two-layer MLP ($\sigma_t \to d \to d$) several orders of magnitude smaller than the ranker; the parameter-matched *Uncertainty Aware Split Attention* ablation (section 5) isolates calibration value from parameter count. γ consumes only σ_t (no forecast, no context features), keeping the

inductive bias clean. Combining the ranker's raw representation with the attention module output treats the ranker's representation as a residual stream. This ResNet like [27] residual architecture is designed to fall back to the base ranker's predictions during periods of high uncertainty, while confidently incorporating forecasted information during periods of low uncertainty. By explicitly accounting for the varying reliability of our forecasts, we prevent the model from learning spurious correlations during sparse data periods and enhance the robustness of the final ranking. Concretely, $\sigma_t \in \mathbb{R}^d$ is the dimension-wise standard deviation across the K TabM [9] ensemble members of their penultimate embeddings at time t, aligning each component of σ_t with the corresponding component of $\mathbf{h}_t$. Any alternative (MC-dropout, deep ensembles, NGBoost) producing a d-dimensional uncertainty vector is a drop-in replacement.

Final architecture: The final architecture (figure 4) consists of a two stage training framework where we first train the forecaster and uncertainty estimator separately. We then freeze the forecaster and uncertainty estimator and train the ranker with the uncertainty scaled attention layer. DART is complementary to advanced architectures for time series forecasting and deep learning ranking models. A stronger ranking backbone or better forecasting/uncertainty strategies/structure can both enhance performance.

5 Experiments

We validate DART on two e-commerce datasets exhibiting distinct drift characteristics. **(1) In-house promotion-ranking dataset:** A large-scale proprietary corpus spanning one year of anonymized, de-identified customer interactions for a single marketplace (14MM data points). Each row is a ⟨customer, product, promotion, date⟩ tuple with a binary redemption target. We use an out-of-time test set (latest 2 months) and an 80:20 train/validation split for the rest. The model consumes 340 features grouped into three entity types: *customer* (RFM engagement scores across 60+ product categories, payment-instrument profile, historical redemption counts), *product* (price, category indicators, active-promotion flags), and *promotion* (discount mechanics, financial partner, eligibility constraints, plus structured attributes extracted from offer text via an LLM and S-BERT [25] embeddings). This dataset exhibits *event-driven* drift with abrupt distribution shifts during promotional events. Because the corpus is non-public, all numbers on this dataset are reported as relative % change versus the baseline ranker, with std propagated via the delta method over 5 runs (different data subsets sampled using different seeds). **(2) Coveo SIGIR [26]:** A public e-commerce benchmark of 36MM browsing events across 4.9MM sessions over 91 days, with 57K products in 174 categories. Sessions map to contexts (x_t), products to artifacts (p_t), and purchase/add-to-cart actions to binary feedback. The forecaster target $g(c, t)$ is daily conversion rate per category. Statistical tests confirm significant drift ($p < 0.005$) in 3 of the 5 largest categories with strong temporal autocorrelation (lag-1 r=0.81). Unlike the in-house data, Coveo exhibits *gradual trend-driven* drift; absolute MRR is reported on this public benchmark.

We use cloud instances with 4 NVIDIA A10G GPUs and 96 vCPU threads. Hyperparameter tuning uses Optuna [1] with 30 trials per model.

Evaluation metric: We use Mean Reciprocal Rank (MRR) (↑) as the headline metric on both datasets. Two properties make it the natural fit for this setting. First, the slate consumed by the customer is short and top-heavy in the e-commerce promotion ranking task: only the first few positions receive meaningful attention, and at most one promotion is redeemed per order; rewarding the rank of the single redeemed (or purchased, on Coveo) artifact and penalising it harmonically with position therefore directly mirrors the observed user behavior more faithfully than rank-insensitive metrics like AUC or position-flat metrics like Precision@k. Second, MRR is well-defined and comparable across both datasets despite their structural differences (per-customer-product slates of variable length on the in-house data, and per-session product orderings on Coveo), because each evaluation instance has exactly one positive-feedback target (redemption / purchase or add-to-cart), which is precisely the regime MRR is built for. We also track MAP and R@1 for completeness on the in-house data; they correlate with MRR and lead to the same model ordering.

Ranking model selection: We compare GBDTs (LightGBM [14] with binary-classification and Learning-to-Rank objectives) against neural tabular models: TabM [9], TabM with piecewise embeddings (TabM-P), and TabNet [3]. TabM-P delivers the best across-the-board performance (+0.12% ± 0.02% MRR, +0.26% ± 0.07% MAP, +1.07% ± 0.03% R@1 vs. GBDT) and is used as the base ranker in subsequent experiments; TabNet underperforms (−1.40% MRR, −1.85% MAP).

Forecaster selection: We evaluate FTT [10], PatchTST [22], DLinear [29], FEDformer [32] and OneNet [31][1], each with multiple regression heads on a shared backbone (one head per financial partner, plus a head forecasting days-to-next-event), using rolling lookback windows $\tau \in \{1, 3, 7, 30\}$ days. Online (test-time-adaptive) variants consistently outperform their offline counterparts on forecaster RMSE, but the picture is less clean on downstream MRR (e.g., PatchTST offline beats online on Overall and Pre-Event), so better point-prediction accuracy does not always imply better ranking. OneNet variants achieve the largest RMSE reductions versus the FTT baseline (−46% to −47% on FP1; −39% on days-to-event). FEDformer benefits least from online adaptation, and on the FP2 head actually shows higher RMSE than the FTT baseline (+92% online, +106% offline), confirming that frequency-domain transformers struggle when the target series has limited periodicity. We use a 3-day forecasting window (δ=3) and TabM for uncertainty estimation throughout (replaceable by other estimators); see section 7 for sensitivity analyses.

Embedding structure matters more than forecast accuracy: Downstream ranking gains are not monotone in forecaster RMSE: TabM-P and OneNet-TCN have near-identical FP1 RMSE but very different MRR (+1.57% vs. +2.23% Overall), and FEDformer's poor FP2 RMSE still produces competitive ranking. DART therefore benefits primarily from the structure of $\mathbf{h}_t$ rather than from point-prediction accuracy, with the forecasting task acting as a pretext objective for temporal group representations; representation quality, not RMSE, is the right selection criterion.

[1]We report two OneNet variants: **OneNet-TCN** uses a Temporal Convolutional Network [5] backbone, while **OneNet** pairs the OneNet ensembling scheme with FSNet [24] as the backbone.

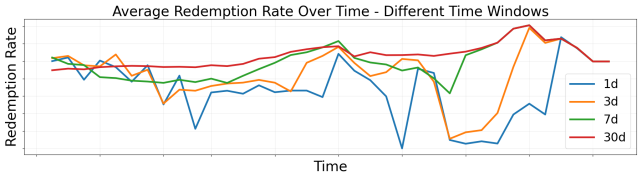

Figure 6: Avg. redemption rate over different time windows.

Temporal Drift Aware Ranking Results: Table 1 compares model variants across regular business-as-usual (BAU) periods, a major promotional event (7 days, Jan 2024, see Fig. 3), and the ±3-day windows around it. For Coveo, we split the test period into High-Drift days ($> 1\sigma$ deviation from the 7-day rolling mean) and Low-Drift days ($\leq 1\sigma$). **Baseline Ranker** is the standard pointwise ranker on binary classification; **Real-time Signal Ranker** is a strong baseline with access to perfect daily redemption rates and event signals; **DART** is our complete model; **DART w/o uncertainty** omits the uncertainty scaling.

We set $\alpha = 0.25$ in Eq. 1 based on empirical analysis (section 7). For fair comparison, BAU metrics on the in-house dataset use a randomly sampled window of equal duration to the event/pre/post-event windows. All DART variants outperform the baseline ranker. The deployable offline configuration (TabM-P) delivers +1.57% ± 0.02% Overall and +11.11%±0.11% during Event, with online OneNet as an upper bound on Event/Pre-Event (+11.99% / +7.94%) and OneNet-TCN leading on Overall (+2.23%); no single online variant dominates all metrics. DART also beats the Real-time Signal Ranker (+11.11% vs. +6.19% Event) despite the latter having perfect current-day rates: the 3-day forecast alone yields +8.12% Event when fed as a direct feature (Table 2), and uncertainty-scaled attention adds a further ~3% on top. Uncertainty-aware attention is essential: DART (TabM-P) at +11.11% Event substantially outperforms the variant without uncertainty at +9.63%. DART also yields +4 to +5% during BAU windows where drift is mild, which we attribute to $\mathbf{h}_t$ carrying steady-state category-level signal (baseline popularity, weak seasonality, group-level demand priors) complementing the ranker. On Coveo, the best DART variant achieves +16.6% ± 0.5% Overall, with near-uniform per-regime lift (+22.0% ± 1.0% High-Drift vs. +22.1% ± 0.1% Low-Drift), showing dual-regime robustness rather than drift-specific gain.

Ablation studies: We conduct two ablation studies to address our research questions. To address **RQ1** (effective integration of forecasting information), we compare alternative ways of feeding forecasting information to the ranker (Table 2). Passing historical redemption rates over rolling windows ($\tau \in \{1, 3, 7, 30\}$ days) as direct features to TabM-P serves as a recency-based baseline analogous to sliding-window adaptation [12]: it improves overall MRR by +0.54% ± 0.10% but Pre-Event MRR *drops* by −1.63% ± 0.10% (Post-Event −3.52% ± 0.06%): recency captures only retrospective patterns, and a static ranker trained mostly on BAU data overfits to a relationship that reverses during transitions. Providing the forecaster's output or embedding as inputs performs better, but both fall short of DART's attention-based integration which additionally benefits from uncertainty scaling.

Table 2: Relative MRR change (↑) of rankers with historical RR features, forecaster embedding and forecaster output as direct inputs, versus the baseline ranker. All use TabM-P as ranker. Std propagated over 5 runs.

Ranker variant	Overall	Event	Pre-Evt	Post-Evt	BAU
+ historical RR	+0.54% ± 0.10%	+7.95% ± 0.05%	-1.63% ± 0.10%	-3.52% ± 0.06%	+1.28% ± 0.10%
+ forecaster output	+0.63% ± 0.03%	+8.12% ± 0.08%	+7.46% ± 0.11%	+0.52% ± 0.12%	+3.92% ± 0.12%
+ forecaster embedding	+0.68% ± 0.02%	+8.25% ± 0.04%	+7.23% ± 0.11%	-0.25% ± 0.06%	+3.49% ± 0.04%
DART (attention + uncertainty)	**+1.57% ± 0.02%**	**+11.11% ± 0.11%**	**+7.37% ± 0.11%**	**+1.85% ± 0.07%**	**+4.60% ± 0.03%**

To address **RQ2** on the effectiveness of uncertainty as a mechanism for handling data variability, Table 3 compares three uncertainty integration strategies. The results confirm that uncertainty integration significantly improves performance beyond a vanilla attention layer, with our scaled-attention design delivering the strongest results. **Uncertainty Scaled Attention** applies a learned projection $\gamma(\sigma_t)$ (a small two-layer MLP, see section 4) to the per-dimension uncertainty vector, and uses its inverse to scale the attention-weighted representation as in Eq. 1. This directly modulates the influence of the forecaster on the ranker based on prediction confidence: when σ_t is large the attention contribution is suppressed and the residual ranker stream dominates; when σ_t is small the forecasted signal is allowed through largely unmodified. **Uncertainty Aware Split Attention** divides the TabM ensemble into two halves. Attention-based forecaster integration is applied to only one half of the ensemble's outputs, while the other half's raw outputs are preserved, and the final score is a learned convex combination $\lambda \cdot \text{Ensemble}_1 + (1-\lambda) \cdot \text{Att}(\text{Ensemble}_2, \mathbf{h}_t, \mathbf{h}_t)$. This creates a partially uncertainty-gated pathway whose parameter count is comparable to scaled attention, which lets us isolate whether the gains from scaled attention come from calibration or simply from added capacity. Empirically the calibration mechanism wins. **Temperature Controlled Attention** uses σ_t to adjust the softmax temperature inside the attention layer:

$$\text{Att}_T(q, K) = \text{softmax}\big(qK^\top / [(1 + \gamma\,\sigma_t)\sqrt{d}]\big)V \tag{2}$$

with higher uncertainty producing a more diffuse attention distribution and lower uncertainty producing a sharper one. Conceptually attractive, but in our experiments it under-performs scaled attention, particularly during transition periods.

Scaled attention consistently outperforms the alternatives across all time periods, with the largest gap during Events and adjacent transition periods where data variability is highest. This supports the conclusion that direct, calibrated scaling of the attention output is the most effective integration mechanism: it lets the model smoothly transition between leaning on the base ranker during high-uncertainty periods and incorporating the forecasted signal when confidence is high.

Table 3: Relative MRR change (↑) of uncertainty integration methods versus the baseline ranker on the in-house dataset. All use TabM-P as ranker, forecaster and uncertainty estimator. Std propagated over 5 runs.

Method	Overall	Event	Pre-Evt	Post-Evt	BAU
Scaled attn.	**+1.57% ± 0.02%**	**+11.11% ± 0.11%**	**+7.37% ± 0.11%**	**+1.85% ± 0.07%**	**+4.60% ± 0.03%**
Split attn.	+0.37% ± 0.02%	+9.48% ± 0.09%	+7.30% ± 0.11%	-0.44% ± 0.13%	+3.82% ± 0.05%
Temp. ctrl. attn.	+0.68% ± 0.02%	+7.92% ± 0.04%	+4.61% ± 0.12%	+1.06% ± 0.07%	+4.30% ± 0.10%

Table 1: Performance comparison across DART configurations on both datasets (MRR ↑). In-house results are during a promotional event in Jan 2024 and are reported as relative % change versus the baseline ranker (std propagated over 5 runs). Coveo results use an out-of-time test split (last 14 days); High/Low-Drift are days where category-level conversion rate deviates by $> 1\sigma$ / $\leq 1\sigma$ from the rolling mean. All use TabM-P as ranker and uncertainty estimator. Results averaged over 5 runs.

		In-house Dataset (relative % vs baseline)					Coveo SIGIR (absolute MRR)		
Model	**Online**	**Overall**	**Event**	**Pre-Evt**	**Post-Evt**	**BAU**	**Overall**	**High-Drift**	**Low-Drift**
Baseline Ranker	✗	—	—	—	—	—	0.1629 ± 0.0004	0.1319 ± 0.0009	0.1722 ± 0.0001
Real-time Signal	✗	+0.09% ± 0.09%	+6.19% ± 0.04%	+2.53% ± 0.11%	+1.32% ± 0.07%	+4.02% ± 0.03%	0.1689 ± 0.0005	0.1520 ± 0.0002	0.1840 ± 0.0007
DART with Different Forecasting Models									
TabM-P w/o unc.	✗	+0.85% ± 0.02%	+9.63% ± 0.10%	+7.04% ± 0.11%	+0.67% ± 0.08%	+4.49% ± 0.04%	0.1688 ± 0.0001	0.1430 ± 0.0005	0.1820 ± 0.0001
TabM-P	✗	+1.57% ± 0.02%	+11.11% ± 0.11%	+7.37% ± 0.11%	+1.85% ± 0.07%	+4.60% ± 0.03%	0.1714 ± 0.0001	0.1522 ± 0.0008	0.1901 ± 0.0001
PatchTST	✗	+1.43% ± 0.02%	+6.48% ± 0.13%	+7.48% ± 0.11%	-0.33% ± 0.07%	+3.49% ± 0.09%	0.1711 ± 0.0001	0.1515 ± 0.0001	0.1948 ± 0.0002
	✓	+1.27% ± 0.03%	+8.12% ± 0.04%	+6.68% ± 0.15%	+1.59% ± 0.06%	+2.62% ± 0.06%	0.1720 ± 0.0001	0.1546 ± 0.0007	0.1929 ± 0.0009
DLinear	✗	+1.06% ± 0.02%	+9.63% ± 0.05%	+7.48% ± 0.12%	+2.50% ± 0.13%	+2.97% ± 0.09%	0.1701 ± 0.0003	0.1529 ± 0.0002	0.1892 ± 0.0001
	✓	+1.98% ± 0.02%	+11.25% ± 0.13%	+7.90% ± 0.11%	+1.75% ± 0.08%	**+5.40% ± 0.03%**	0.1755 ± 0.0005	0.1511 ± 0.0006	0.1983 ± 0.0004
FEDformer	✗	+1.32% ± 0.02%	+7.54% ± 0.05%	+5.28% ± 0.12%	+0.48% ± 0.11%	+5.13% ± 0.04%	0.1752 ± 0.0009	0.1501 ± 0.0004	0.1920 ± 0.0008
	✓	+1.75% ± 0.07%	+8.98% ± 0.09%	+6.98% ± 0.15%	+2.36% ± 0.07%	+5.09% ± 0.07%	0.1760 ± 0.0002	0.1582 ± 0.0003	0.2001 ± 0.0008
OneNet-TCN	✓	**+2.23% ± 0.02%**	+11.36% ± 0.09%	+7.45% ± 0.13%	**+2.98% ± 0.07%**	+4.48% ± 0.11%	0.1822 ± 0.0003	0.1595 ± 0.0007	0.2033 ± 0.0002
OneNet	✓	+2.16% ± 0.09%	**+11.99% ± 0.05%**	**+7.94% ± 0.15%**	+1.75% ± 0.07%	+4.68% ± 0.09%	**0.1899 ± 0.0007**	**0.1609 ± 0.0008**	**0.2102 ± 0.0001**

6 Deployment and Productionization

We have validated DART against a production promotion-ranking system across multiple model iterations. Real-time serving is required: pre-computing a ranked list for every customer-product pair is infeasible at the scale of millions of customers and millions of products, so the model is invoked synchronously when a customer lands on a product page. The ranking-service latency budget is ∼10ms, which rules out heavy transformer rankers without auto-scaling GPU fleets whose cost rarely justifies the incremental ranking accuracy.

Online forecasters are particularly hard to deploy in this domain because (i) the model is invoked millions of times per second, making per-request weight updates infeasible without aggressive sampling, and (ii) ground-truth redemption signal is delayed by 1–2 days because successful offer attach is confirmed by downstream payment systems that account for cancellations and reversals. Online models also demand higher compute due to combined training and inference resources. The trade-off between cost and accuracy thus favors offline forecasters in this setting.

DART's two-stage design is well-suited to these constraints: only the lightweight ranker needs to be hosted on the real-time path, while the forecaster and uncertainty estimator can run in batch on a daily cadence with embeddings cached at the date-and-group level (small storage footprint). At inference, the relevant cached embedding can be fetched by date and combined with the ranker output via the learned attention layer. The two components can be retrained on independent schedules: typically the ranker monthly (or as business needs dictate) and the forecaster daily. A month-long customer-randomized A/B test of personalized ranking models against heuristic alternatives and baselines yield approximately **+6**% relative MRR gains online ($p < 0.05$), confirming offline gains translate to production lift.

7 Hyperparameter Sensitivity

We analyse two key hyperparameters in DART: the residual weight α in Eq. 1, which balances the original ranker output against the uncertainty-scaled attention output, and the forecasting window δ, which controls how far ahead the forecaster predicts.

Residual weight α: We sweep $\alpha \in \{0.0, 0.1, 0.25, 0.5, 0.75, 0.9, 1.0\}$, where $\alpha = 0$ relies entirely on the attention-modified output and $\alpha = 1$ collapses to the baseline ranker. Table 4 reports relative MRR change versus the $\alpha = 1$ baseline. Pure attention integration ($\alpha = 0$) underperforms because the attention output cannot fully substitute for the fine-grained context-promotion signal in the ranker. The best overall performance occurs at $\alpha = 0.25$, with the model giving 75% weight to the attention-modified output while retaining a residual to the original ranker. Beyond $\alpha = 0.5$, performance steadily declines toward the baseline; the gap is most pronounced during Event periods where temporal drift is most severe. Pre-Event performance is comparatively stable over $\alpha \in [0.25, 0.5]$, suggesting forecasted signals are particularly reliable during anticipatory periods. We use $\alpha = 0.25$ in all main results.

Table 4: Sensitivity to residual weight α (relative MRR change vs. baseline). Best in bold, second best underlined.

α	Overall	Event	Pre-Evt	Post-Evt	BAU
0.00	+0.50%	+8.26%	+5.73%	-0.17%	+3.32%
0.10	+1.18%	+10.16%	+6.83%	+1.00%	+4.09%
0.25	**+1.57%**	**+11.11%**	<u>+7.37%</u>	**+1.85%**	**+4.60%**
0.50	<u>+1.38%</u>	<u>+10.51%</u>	**+7.65%**	<u>+1.39%</u>	<u>+4.31%</u>
0.75	+0.88%	+7.44%	+5.44%	+0.46%	+3.49%
0.90	+0.38%	+3.41%	+2.68%	-0.07%	+1.99%
1.00 (baseline)	0.00%	0.00%	0.00%	0.00%	0.00%

Forecasting window δ: We sweep $\delta \in \{1, 3, 7, 14, 30\}$ days and track both forecaster RMSE (relative to the FTT baseline) and downstream ranker MRR (relative to the baseline ranker). Table 5 shows that $\delta = 3$ days is jointly optimal across both metrics. Three factors drive this: (i) shorter windows ($\delta = 1$) suffer from high-frequency daily noise (figure 6); (ii) major promotional events typically span 3–7 days, and financial partners activate event-specific offers 1–3 days before and deactivate them 1–2 days after, so a 3-day window

aligns naturally with these transitions; (iii) longer horizons carry higher forecast uncertainty, which DART's attention mechanism then down-weights, eroding the benefit of looking further ahead. The Event and Pre-Event metrics are most sensitive to window choice, confirming that temporal alignment matters most during high-drift periods.

Table 5: Forecasting window δ vs. forecaster RMSE (relative to FTT baseline; negative is improvement) and ranker MRR (relative to baseline ranker). Results use the TabM-P forecaster with uncertainty-scaled attention.

δ	Forecaster RMSE rel. change			Ranker MRR rel. change		
	FP1	FP2	Days	Overall	Event	Pre-Evt
1 day	-35.94%	-13.89%	+8.25%	+0.84%	+7.75%	+5.03%
3 days	**-46.88%**	**-36.11%**	**-38.90%**	**+1.57%**	**+11.11%**	**+7.37%**
7 days	-44.53%	-27.78%	-33.40%	+1.20%	+9.82%	+6.68%
14 days	-38.28%	-5.56%	-22.41%	+0.75%	+7.39%	+4.62%
30 days	-26.56%	+25.00%	-10.57%	+0.38%	+3.59%	+2.68%

8 Generalisability beyond Ranking

Although DART is motivated by ranking under temporal drift, its uncertainty-scaled attention is agnostic to the downstream task. To demonstrate this, we apply DART to tabular regression on the Shifts Weather Prediction benchmark [20, 21], a public dataset of ~10MM meteorological measurements across locations and times where the task is short-horizon temperature prediction. As noted by Malinin et al. [21], this dataset shares structural challenges with medical and financial forecasting: heterogeneous features, non-uniform sample distribution, and significant domain drift. We aggregate measurements at a location-date level (10km regions) and form rolling-window features (1h, 6h, 1d, 3d, 7d, 14d, 30d) of meteorological variables and historical temperatures to predict the next 3-day average temperature. We use TabM as the ranker (now a regressor) and the uncertainty estimator, and apply DART on top of FTT, PatchTST, TabM and FEDformer forecasters.

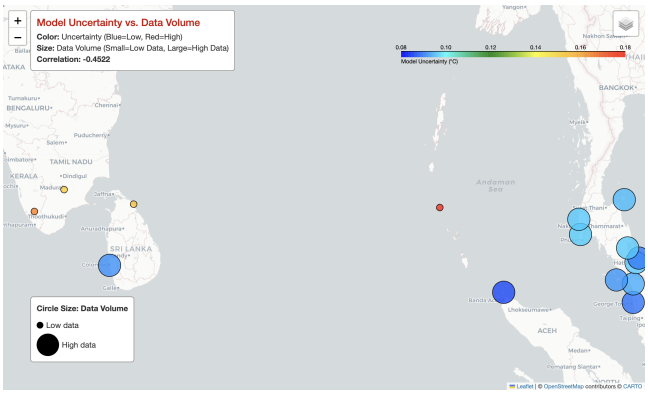

Figure 7: Geographic map of model uncertainty for the temperature regression task. Regions with sparse observations exhibit higher uncertainty.

Figure 7 shows that estimated model uncertainty is negatively correlated with observation density, mirroring the data-availability pattern we observed in the promotion-ranking setting and validating that the uncertainty signal carries the same physical interpretation across very different domains. Across all evaluated backbones, DART reduces RMSE versus the corresponding base model: relative RMSE reductions are **4.3**% for TabM, **7.1**% for FTT, **2.6**% for PatchTST, and **4.9**% for FEDformer; the best DART variant reaches RMSE **1.4288** (~**4.9**% better than its FEDformer base and **6.8**% better than the TabM-only baseline), with R^2 improving up to **0.851**. Critically, DART benefits hold under both gradual seasonal drift and abrupt temperature transitions that the vanilla regressor mishandles, confirming that uncertainty-scaled attention provides a domain-agnostic mechanism for adapting to temporal drift without architecture changes specific to the downstream task.

9 Conclusion and Future Work

In this paper, we address the problem of temporal drift in artifacts in the task of ranking with focus on the e-commerce domain. Our primary contribution, DART, is a novel two-stage architecture that combines a multi-task forecaster with a ranker through an uncertainty-aware attention mechanism, enabling effective offline adaptation to temporal shifts. Through comprehensive ablation studies, we found that attention-based integration of forecasting embeddings significantly outperforms direct feature inclusion approaches. One of the limitations of our work is that we do not address the drift in context $\mathcal{X}_t$ over time since it also relates to the problem of "cold-start" which can be a future extension to our work.

Acknowledgments

We thank Srujana Merugu for her fruitful comments, corrections and inspiration that helped us crystallize the problem statement and widen the impact of this work.

References

[1] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A Next-generation Hyperparameter Optimization Framework. arXiv:1907.10902 [cs.LG] https://arxiv.org/abs/1907.10902

[2] Oren Anava, Elad Hazan, Shie Mannor, and Ohad Shamir. 2013. Online learning for time series prediction. In *Conference on learning theory*. PMLR, 172–184.

[3] Sercan O. Arik and Tomas Pfister. 2020. TabNet: Attentive Interpretable Tabular Learning. arXiv:1908.07442 [cs.LG] https://arxiv.org/abs/1908.07442

[4] Manuel Baena-Garcıa, José del Campo-Ávila, Raul Fidalgo, Albert Bifet, Ricard Gavalda, and Rafael Morales-Bueno. 2006. Early drift detection method. In *Fourth international workshop on knowledge discovery from data streams*, Vol. 6. 77–86.

[5] Shaojie Bai, J Zico Kolter, and Vladlen Koltun. 2018. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271* (2018).

[6] Michiel de Jong, Yury Zemlyanskiy, Joshua Ainslie, Nicholas FitzGerald, Sumit Sanghai, Fei Sha, and William Cohen. 2023. FiDO: Fusion-in-Decoder optimized for stronger performance and faster inference. arXiv:2212.08153 [cs.CL] https://arxiv.org/abs/2212.08153

[7] Siddartha Devic, Aleksandra Korolova, David Kempe, and Vatsal Sharan. 2024. Stability and multigroup fairness in ranking with uncertain predictions. *arXiv preprint arXiv:2402.09326* (2024).

[8] Isvani Frías-Blanco, José del Campo-Ávila, Gonzalo Ramos-Jiménez, Rafael Morales-Bueno, Agustín Ortiz-Díaz, and Yailé Caballero-Mota. 2015. Online and Non-Parametric Drift Detection Methods Based on Hoeffding's Bounds. *IEEE Transactions on Knowledge and Data Engineering* 27, 3 (2015), 810–823. doi:10.1109/TKDE.2014.2345382

[9] Yury Gorishniy, Akim Kotelnikov, and Artem Babenko. 2025. TabM: Advancing Tabular Deep Learning with Parameter-Efficient Ensembling. arXiv:2410.24210 [cs.LG] https://arxiv.org/abs/2410.24210

[10] Yury Gorishniy, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. 2021. Revisiting deep learning models for tabular data. *Advances in neural information processing systems* 34 (2021), 18932–18943.
[11] Maria Heuss, Daniel Cohen, Masoud Mansoury, Maarten de Rijke, and Carsten Eickhoff. 2023. Predictive uncertainty-based bias mitigation in ranking. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. 762–772.
[12] Rolf Jagerman, Ilya Markov, and Maarten de Rijke. 2019. When people change their mind: Off-policy evaluation in non-stationary recommendation environments. In *Proceedings of the twelfth ACM international conference on web search and data mining*. 447–455.
[13] Heegon Jin, Seonil Son, Jemin Park, Youngseok Kim, Hyungjong Noh, and Yeonsoo Lee. 2024. Align-to-Distill: Trainable Attention Alignment for Knowledge Distillation in Neural Machine Translation. arXiv:2403.01479 [cs.CL] https://arxiv.org/abs/2403.01479
[14] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems* 30 (2017).
[15] Alexander C. Li, Yuandong Tian, Beidi Chen, Deepak Pathak, and Xinlei Chen. 2024. On the Surprising Effectiveness of Attention Transfer for Vision Transformers. arXiv:2411.09702 [cs.LG] https://arxiv.org/abs/2411.09702
[16] Xin Li, Dongze Lian, Zhihe Lu, Jiawang Bai, Zhibo Chen, and Xinchao Wang. 2023. GraphAdapter: Tuning Vision-Language Models With Dual Knowledge Graph. arXiv:2309.13625 [cs.CV] https://arxiv.org/abs/2309.13625
[17] Haonan Lin, Wenbin An, Jiahao Wang, Yan Chen, Feng Tian, Mengmeng Wang, Guang Dai, Qianying Wang, and Jingdong Wang. 2024. Flipped Classroom: Aligning Teacher Attention with Student in Generalized Category Discovery. In *Advances in Neural Information Processing Systems*, A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (Eds.), Vol. 37. Curran Associates, Inc., 60897–60935. doi:10.52202/079017-1947
[18] Chenghao Liu, Steven CH Hoi, Peilin Zhao, and Jianling Sun. 2016. Online arima algorithms for time series prediction. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 30.
[19] Pengqian Lu, Jie Lu, Anjin Liu, and Guangquan Zhang. 2025. Early concept drift detection via prediction uncertainty. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 39. 19124–19132.
[20] Andrey Malinin, Andreas Athanasopoulos, Muhamed Barakovic, Meritxell Bach Cuadra, Mark JF Gales, Cristina Granziera, Mara Graziani, Nikolay Kartashev, Konstantinos Kyriakopoulos, Po-Jui Lu, et al. 2022. Shifts 2.0: Extending the dataset of real distributional shifts. *arXiv preprint arXiv:2206.15407* (2022).
[21] Andrey Malinin, Neil Band, German Chesnokov, Yarin Gal, Mark JF Gales, Alexey Noskov, Andrey Ploskonosov, Liudmila Prokhorenkova, Ivan Provilkov, Vatsal Raina, et al. 2021. Shifts: A dataset of real distributional shift across multiple large-scale tasks. *arXiv preprint arXiv:2107.07455* (2021).
[22] Yuqi Nie, Nam H Nguyen, Phanwadee Sinthong, and Jayant Kalagnanam. 2022. A time series is worth 64 words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730* (2022).
[23] Geon Yeong Park, Jeongsol Kim, Beomsu Kim, Sang Wan Lee, and Jong Chul Ye. 2023. Energy-Based Cross Attention for Bayesian Context Update in Text-to-Image Diffusion Models. arXiv:2306.09869 [cs.CV] https://arxiv.org/abs/2306.09869
[24] Quang Pham, Chenghao Liu, Doyen Sahoo, and Steven CH Hoi. 2022. Learning fast and slow for online time series forecasting. *arXiv preprint arXiv:2202.11672* (2022).
[25] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. https://arxiv.org/abs/1908.10084
[26] Jacopo Tagliabue, Ciro Greco, Jean-Francis Roy, Federico Bianchi, Giovanni Cassani, Bingqing Yu, and Patrick John Chia. 2021. SIGIR 2021 E-Commerce Workshop Data Challenge. In *SIGIR eCom 2021*.
[27] Andreas Veit, Michael J Wilber, and Serge Belongie. 2016. Residual networks behave like ensembles of relatively shallow networks. *Advances in neural information processing systems* 29 (2016).
[28] Tao Yang, Cuize Han, Chen Luo, Parth Gupta, Jeff M Phillips, and Qingyao Ai. 2024. Mitigating exploitation bias in learning to rank with an uncertainty-aware empirical Bayes approach. In *Proceedings of the ACM Web Conference 2024*. 1486–1496.
[29] Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. 2023. Are transformers effective for time series forecasting?. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 37. 11121–11128.
[30] YiFan Zhang, Weiqi Chen, Zhaoyang Zhu, Dalin Qin, Liang Sun, Xue Wang, Qingsong Wen, Zhang Zhang, Liang Wang, and Rong Jin. 2024. Addressing Concept Shift in Online Time Series Forecasting: Detect-then-Adapt. *arXiv preprint arXiv:2403.14949* (2024).
[31] Yi-Fan Zhang, Qingsong Wen, Xue Wang, Weiqi Chen, Liang Sun, Zhang Zhang, Liang Wang, Rong Jin, and Tieniu Tan. 2023. OneNet: Enhancing Time Series Forecasting Models under Concept Drift by Online Ensembling. arXiv:2309.12659 [cs.LG] https://arxiv.org/abs/2309.12659
[32] Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. 2022. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. In *International conference on machine learning*. PMLR, 27268–27286.
[33] Jianhan Zhu, Jun Wang, Michael Taylor, and Ingemar J Cox. 2009. Risk-aware information retrieval. In *European Conference on Information Retrieval*. Springer, 17–28.

A Appendix

A.1 GenAI Usage Disclosure

Popular SOTA generative AI tools were used for minor rephrasing and editing of the text in this publication. Generative AI tools have NOT been used in the ideation, experimentation, result tabulation and plot generation phase, they have only been used for light editing of the text to improve the language and structure of the publication. The authors take full responsibility and ownership of the method proposed and its results.