We thank all the reviewers for their valueable suggestions. Typos will be fixed and related works will be revised.

R1Q1: No DIEN results on product dataset. Necessary to give reasons.

R1A1: In preliminary studies, DIEN performs worse than Transformer on product dataset (consists with Tab1). We did not report the exact number because DIEN is not feasible given the high latency in our CPU-based online system (Fig2).

R2Q1: Claim "KFAtt predict interests when few historical behaviors are relevant, Line53" lacks experimental support.

R2A1: Line256-258&264 and Tab1&2. We construct a more challenging test subset "*New*", where *NO* historical behaviors are relevant to the current query. On "New", KFAtt achieves *larger* performance gain compared to that on "All", directly supporting the claim. A study of history length is helpful, but not essential given results on "New".

R2Q2: The implementation details are in a session-based CTR prediction, not reasonable to use long history.

R2A2: Our system is NOT in *session-based* CTR prediction. Long behaviors of identities are totally traced. Our problem setting is *user behavior modeling* [23,24,25] (*Our Title*), i.e., to predict user's current interest from rich historical behaviors, Line26. We apologize for that this misunderstanding might come from a reference compiling error. Line189 [11] should be *Feng Yufei. Deep session interest network for click-through rate prediction.* "Session" here means a partition method in processing very long behavior sequences, irrelevant to "Session"-based CTR prediction.

R2Q3: How an accurate estimation of $\boldsymbol{\mu}_q$, σ_q σ'_m from a 2-layer MLP with only p/k as input? Ground truth available?

R2A3: Only embeddings $\mathbf{q}, \mathbf{k}$ are needed for $\boldsymbol{\mu}_q, \sigma_q, \sigma'_m$ (Line127&159). No ground truth. The intuitive reason for this simple but accurate estimation is that they are trained and shared across a great many users with the same query.

R2Q4: The semantics of $\boldsymbol{\mu}_q$? And what kind of distance measure is used to quantify the value of θ_t using $k_t$and q?

R2A4: $\boldsymbol{\mu}_q$ is Gaussian distribution mean, namely the mean interest under same $\mathbf{q}$ across all users (Line122-124), NOT avg-pool of user embedding. Calculated by $\boldsymbol{\mu}_q = MLP(\mathbf{q})$ (Line127). Distance measured as Line200.

R2Q5: Reproducibility. **R2A5**: #heads=4. Others follow codes of DIEN. Will consider open source upon acceptance.

R3Q1: How Kalman Filtering (KF) provides new insights, given the simple solutions to the two problems?

R3A1: Although useful in sequential scenarios, KF is essentially a *sensor-fusion* method. The fusion is estimated by MAP, whose solution is a weighted-sum of *prior* and sensor measurements. Similarly in behavior modeling, each historical behavior can be considered as a measurement of the current interest. So the current interest is also a fusion, which is naturally under MAP framework and thus fits KF. While conventional attentions (*expectation*) neglect query priors and thus suffer from cold start. Moreover, KFAtt is far more than "2 simple modifications". With the additional σ_q and σ'_m, it assigns stronger prior and capping to specific queries than to general ones, Line125. To see this superior, we now show AUC gain from "simply including query-specific prior, $\sigma_q = 1$" to KFAtt-b, and gain from "simply frequency capping" to KFAtt-f: *All*: +0.0026 | +0.0035 ||| *New*: +0.0037 | +0.0046 ||| *Infreq*: +0.0011 | +0.0025.

R3Q2: Whether it is nontrivial improvements. Is a +4.4% CTR gain big or small?

R3A2: Our base is highly optimized (400M users Tab2 Supp), CTR+4.4% => Income+$0.1Billion/year, Big Gain.

R3Q3: Different implementation choices (e.g. $\boldsymbol{\mu}_q$) spread across the paper.

R3A3: Query mean and std ($\boldsymbol{\mu}_q, \sigma_q, \sigma'_m$) are from MLP (Line127&159). Distance σ_t, σ_m are as Line200. This is the only final implementation. Other variants are only for ablation studies of adaption to other attentions (Tab2).

R3Q4: If $\exp(\mathbf{q}^T W_Q W_K \mathbf{k}_m)$ is used as $1/\sigma_m^2$, how is KFAtt-f calculated? Numerically stable?

R3A4: In Eq(10), $\frac{1}{\sigma_m^2 + \sigma_m'^2/n_m} = \frac{1/\sigma_m^2}{1+1/\sigma_m^2 \cdot \sigma_m'^2/n_m} = \frac{\exp(\mathbf{q}^\top W_Q W_K \mathbf{k}_m)}{1+\exp(\mathbf{q}^\top W_Q W_K \mathbf{k}_m) MLP(\mathbf{k}_m)/n_m}$. Numerically stable.

R4Q1: Is global prior equivalent to including an "entry" weighed by 1? | **R4A1**: No. Pls see R3A1 for details.

R4Q2: Sensitivity analysis to de-duplication algorithm.

R4A2: Amazon dataset contains 3 levels of categories. We now show AUC gain from using 3rd level de-duplication (as in paper) to 2nd, and gain from 3rd to 1st. *All*: -0.0014 | -0.0023 ||| *New*: -0.0019 | -0.0022 ||| *Infreq*: +0.0008 | +0.0010. Coarser de-duplications benefit queries from infrequent cates but harm frequent ones, leading to lower performance on *All*. KFAtt-f with any level of de-duplications outperforms KFAtt-b and other STOAs, not that sensitive.

R4Q3: Swapping the values learned by KFAtt to Vanilla and other models is wrong.

R4A3: Thank reviewer's help in finding an ambiguous description. Line272 should be "we assign attention weights calculated by Vanilla, DIN and Transformer to $1/\sigma_t^2$ and $1/\sigma_m^2$ in Eq (6,10)". Namely, we plug KFAtt to these attentions and show consistent improvements brought to them.