- On reviewers’ concern about experimental results

When we demonstrate the experimental results, we focus on the comparison aspect to show the advantage of involving contextual information and position discounts over alternative methods. Actually the learning curves will show a good concave/convex shape when n, the number of rounds, is small. Taking Figure 1(b) for an example, the learning curves show a good concave shape when n <= 2000, during which the cosine similarity between \theta\_\* and its estimate \hat{\theta} increases to 0.995. This means that our algorithm learns quickly, which is because we use a small d=10 as the dimension of the latent vector \theta\_\*, making the learning task easy. When n>2000, due to the randomness, the accuracy of $\hat{\theta}$ increases very slowly. Thus there is a very small probability that $\hat{\theta}$ will lead to a wrong choice of the action. Such a small probability is decreasing, but in a very slow rate, which results in the almost linear shape of the curves when n is relatively large. We will test harder learning tasks with larger d values to make the learning curves exhibit more obvious concave shape.

- On reviewer 1’s comment “the authors never give an example of a more general reward function or experiment with it.”, and the request to clarify Sec. 4.3 and Sec. 5.3

We indeed provide an example in the contextual bandit setting that cannot be covered if we simply extend existing cascading bandit results to involve contextual information. The example is given in Section 4.3 and is experimented in Section 5.3 with Figure 3(c). Perhaps our original explanation is unclear, so we provide further details here.

This example considers a network routing problem with edge latency. The latency of each edge follows an exponential distribution with a cutoff value, say 2\tau. That is, after time 2\tau, we do not wait for the actual delay feedback and just use 2\tau as the feedback. The cutoff exponential distribution has bounded support and thus satisfies the R-sub-Gaussian property. We regard an edge as blocked if its latency is larger than threshold \tau (say as an application requirement). An action is a routing path from the source to the target. The reward of the action is 1 if no edge on the path is blocked, and 0 if some edge on the path is blocked (ignoring position discount in this example). After an action is played in each round, the observed feedback is the delays of edges up until the first blocked edge (for the first blocked edge, we still give it a chance after \tau up to 2\tau to observe its delay). The mean delay of an edge e with the cutoff exponential distribution is determined by the linear combination of the context of the round x\_{t,e} and the latent vector \theta\_\*.

In this example, we can use observed delays and contexts to apply linear regression to estimate \theta\_\*, as given in the paper. However, the expected reward function as a function of mean delays of edges is a complicated function (even without a closed-form formula): we need to transform the mean delays of the cutoff exponential distribution to the non-cutoff exponential distribution, and then use the latter to derive the blocking probability with threshold \tau. Thus it is not covered by existing cascading bandit studies. Yet we can still argue that this function is monotone and Lipschitz continuous, and thus obtain regret bound from Theorem 4.3. On the other hand, if we treat edge blocked or unblocked as the feedback in order to fit into the existing cascading bandit framework, we cannot apply linear regression to learn the latent vector \theta\_\*, because the probability of edge blocking or nonblocking is not a linear combination of contexts and \theta\_\*.

This example is also experimented in Section 5.3 with Figure 3(c).

- On reviewer 1’s concern about 1/p\* being arbitrarily large and p\* not tied to the reward function.

Our understanding is that p\* is always tied to the reward function. If not, consider an extreme example, where a sequence of base arms has a tiny probability p observing the last base arm in the sequence, making p\* very small and 1/p\* very large. If the outcome of the last base arm is not tied to the reward function, it means that observing the last base arm or not does not make a difference, then we can remove it from the sequence so that p\* is not defined based on this observation probability. In general, if observing a base arm is not tied with the reward function, we can remove it in the problem instance so that it does not affect p\*. When observing a base arm indeed could make a difference in the final reward and the optimal action selection, then one has to observe this base arm, no matter how small the observation probability is, and in this case a regret with 1/p\* is reasonable and inevitable. We will add proper discussions in the paper to clarify this point.

- On reviewer 1’s concern about monotonicity assumption

This assumption implies if the recommended list has higher reward means, then the expected reward of the list should also get higher. We think this is a relatively basic assumption.

- On reviewer 1’s concern about last paragraph in Section 5.3

This experiment is to demonstrate that the knowledge of position discount \gamma could be important in the learning process. For the network routing tests, Fig. 3(d) shows that without considering position discount (\gamma = 1), the regret is much larger than using a position discount close to the true value of (\gamma\*= 0.9). The intuition is that a position discount will guide the selection towards selecting paths that are likely to be reliable initially, while no position discount will guide the selection towards globally reliable paths. Essentially, in the network routing case different position discounts lead to different optimal solutions. This is not the case for the disjunctive cascading bandit (selecting top k base arms with the highest probability of one of them returning 1) --- with all position discounts, the top k choices are the same. We just want to illustrate the importance of position discounts in some applications.

- On reviewer 4’s concern about Line 657-663

This part is to justify the additional term in the regret due to accommodating contextual information. Our result is based on the linear bandits (Abbasi-Yadkori et al. 2011), which also introduces an additional term compared to the classical MAB.

- On reviewer 4’s concern about the context in Section 5.3

We generate the contextual information x\_{t,a} randomly. CombCascade performs much worse because they do not use the contextual information. Even though the context is random, it still guides C^3-UCB to learn the latent \theta\_\* much better, so that given a new (and random) context, C^3-UCB knows much better what action to choose. Without the use of contexts, CombCascade can only select the action that performs well for the \*average\* context, which is certainly much worse than C^3-UCB, which can perform well on \*every\* context after the learning period.

- On reviewers’ other concerns

The V\_t norm in line 376 is defined in line 362-364.

The simple path in Line 812 is a directed path without cycles.