

1 We thank all the reviewers for their comments and feedback.

2 First, we would like to address Reviewer #3's concerns about the quality of the samples generated by our algorithm.
3 In this paper, we propose an efficient algorithm for sampling from an established family of distributions that select
4 representative samples of data: determinantal point processes of size k , a.k.a., k -DPPs. The effectiveness of k -DPPs at
5 producing representative samples has been shown by extensive prior work at top ML conferences (see the first paragraph
6 of Section 1). In our paper, we formally prove that our algorithm samples *exactly from the target k -DPP distribution*.
7 Thus, the quality of the samples produced by the algorithm is established by all the prior work on k -DPPs.

8 To Reviewer #2

9 *"Table 1 does not discuss that the bound for the proposed algorithm holds with probability $1 - \delta$, while I assume that*
10 *n^3 bound is with probability 1."*

11 Thanks for pointing this out. We will clarify that the eigendecomposition-based methods, although much slower, have a
12 deterministically bounded runtime.

13 *"In Figure 1, the proposed algorithm and DPP-VFX take 50 seconds even with very small n . It would help to discuss*
14 *why these methods takes nontrivial time with very small n ."*

15 The higher complexity at low n is due to the search and rejection
16 steps in the k -DPP sampler. Specifically, α -DPP samples from a
17 k -DPP by repeatedly sampling from a random-size DPP until it
18 generates a sample with size exactly k . Our theory ensures that
19 this will happen after only a small number of rejections. In the
20 experiments from Figure 1 in the paper, this amounts to roughly
21 8 to 10 rejections until acceptance (see Figure 1 in this response).
22 A single DPP sample requires ~ 5 seconds to generate using
23 α -DPP, resulting in a runtime around 50 seconds for the k -DPP.
24 Note however that this is mostly because we do not choose a
25 different oversampling parameter for different sample sizes. In
26 particular, for small n we can increase the oversampling both in
27 the Nyström approximation as well as in the intermediate sample
28 size to reduce the number of rejections and improve overall
29 runtime. We will include this discussion in the final version.

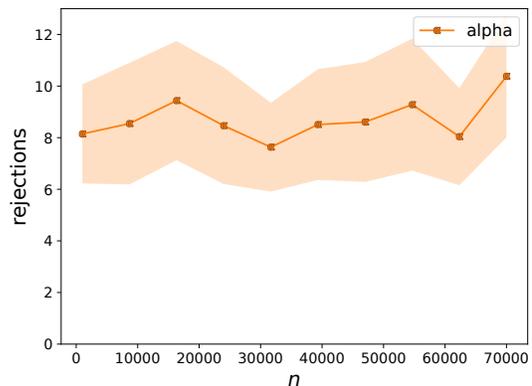


Figure 1: Number of rejections when converting a random-size DPP into a k -DPP.

30 To Reviewer #3

31 *"As mentioned in the abstract, previous naive heuristic method can not guarantee if the selected items are representative.*
32 *How can the proposed method ensure that the selected samples resemble the target distribution?"*

33 Theorem 1 clearly states that the samples returned by α -DPP exactly follow the target (i.e. k -DPP) distribution. That is,
34 we rigorously prove (Lemma 6 in Appendix A) that α -DPP's induced distribution does not simply resemble a k -DPP, or
35 is ε -close like an MCMC sampler, but strictly follows the definition of a k -DPP distribution.

36 *"I think this should be further evaluated in the experiment section. For example, plot figures using algorithm like t-SNE*
37 *or compute the distributions using some statistical metrics."*

38 We formally prove that the two distributions are equal. Any statistical discrepancy we could find by running an
39 experiment would only be a statistical error due to the finite sample size.

40 To Reviewer #4

41 *"...the full sampling should be clearly and concisely presented, preferably in an additional algorithmic or pseudocode*
42 *block."*

43 We will move the paragraph on caching strategy in Appendix E earlier in the appendix, and add more details on the full
44 algorithmic pipeline. Note that in addition we provide in the supplementary material a reference implementation in
45 python that also shows how to integrate the whole algorithm.

46 *"Would it be possible to extend the α -DPP sampling approach presented in this paper to sampling from a standard DPP,*
47 *rather than a k -DPP?"*

48 This can be done, and it is indeed already the case, in that, α -DPP (Algorithm 1) by itself samples from a standard DPP
49 but rescaled by a constant factor, which is then used to construct k -DPP samples. More generally, α -DPP can sample
50 from a standard DPP without looking at all elements whenever it has a prior upper bound on the marginal inclusion
51 probabilities (see "Beyond uniform sampling" on L222). Rescaling provides such a bound for k -DPPs. We hope that, as
52 the usage of α -DPP increases, more structured bounds will be discovered and integrated in the algorithm.