We thank all reviewers for their positive comments.

Reviewer #1 **Learning:** Our tester implicitly learns, too: To learn a DPP, we can run the tester and return any DPP in the candidate set $\mathcal{M}$ that is accepted by the χ^2-ℓ_1 test (if any). By Theorem 1, with high probability, the returned DPP is close to the DPP whose samples we observe. We will make this more explicit. **Lemma 4:** "high probability" means with an arbitrarily large constant probability. We will clarify this. **Uniform distribution:** In Line 286 we state why the uniform measure is a DPP. We will clarify this. **Confidence Interval:** The tester uses $|\hat{K}_{i,j}|$ as an estimate for $|K^*_{i,j}|$ (the sign is harder to estimate). For some parameter u, concentration assures that w.h.p. $|\hat{K}_{i,j}|$ belongs to $(|K^*_{i,j}| - u, |K^*_{i,j}| + u)$. The algorithm picks multiple equally-distant values from the intervals $(|\hat{K}_{i,j}| - u, |\hat{K}_{i,j}| + u)$ and $(-|\hat{K}_{i,j}| - u, -|\hat{K}_{i,j}| + u)$. We will clarify this. **Section 5.2:** The main idea is to use the coupling to bound the acceptance probability of the tester for $\pi_{\bar{z}}(K^*)$ instead of K^*, by applying Lemma 1, and then use their closeness to transfer the result. We will clarify this. **Run time:** is not generally polynomial as $|\mathcal{M}|$ is not polynomially bounded by n. We will add this.
Other comments: We will add/clarify these and correct the typos. Thank you.

Reviewer #2 We will add synthetic data experiments in the supplement. As the reviewer correctly pointed out, our exponential lower bound leaves no hope for a more efficient tester. But we hope that it motivates follow-up works studying structural assumptions, sub- and super-classes that may allow better results. **Reducing the order of $|\mathcal{M}|$:** We are not aware of any systematic way for this. Yet, a simple idea is to consider the identifiability classes of K: for each $i \in [n]$, multiplying the ith row and column of the marginal kernel by -1 does not change the distribution. This defines an equivalence relation between matrices, where each matrix is equivalent to 2^{n-1} others. Currently, we are considering all of the 2^{n-1} matrices of a class in $\mathcal{M}$. Also, as a heuristic method, one can substitute the discrete search in $\mathcal{M}$ by an iterative approach: We are looking for a DPP in $\mathcal{M}$ for which the statistic $Z^{(m)}$ is less than C. One can see $Z^{(m)}$ as a smooth function of the kernel K, and minimize it using Gradient Descent (GD) with initialization $\hat{K}$.

Reviewer #3 **L42:** We will clarify this. **L115:** Indeed, our tester is deterministic. Except the statement of choosing a random distribution in $\mathcal{F}$, all high probability statements, including the one in L115, are w.r.t. q, whose samples we observe. **L148-150:** This is a typo. We meant that using their algorithm results in sub-optimal sample complexity for testing. We will correct this. **L186:** We will add a note on the cardinality of $\mathcal{M}$. **L288:** The "high probability" in L292 refers only to the randomness of selecting h from $\mathcal{F}$. We will explain this. Randomization is necessary for the hardness result: compared to any fixed distribution in $\mathcal{F}$, randomization further decreases the ℓ_1 distance to the uniform measure, which makes testing harder. **Testing continuous DPPs:** testing in continuous space without any parameterization is generally infeasible with finite samples. For example, consider ϵ-uniformity testing over $[0, 1]$: One can divide $[0, 1]$ into N sub-intervals and consider only distributions that assign constant density to each. This discretization transfers the lower bound $\sqrt{N}/\epsilon^2$ of the discrete problem to the continuous case, for any N. We conjecture that a similar negative result holds for testing continuous DPPs. **Approximate Sampler:** We are currently not sure if our results generalize to an approximate sampler. The tester starts by estimating K^* by computing the marginal probabilities for every $i \in [n]$ and $\{i, j\} \subseteq [n]$. An approximate sampler with ℓ_1 error e adds an additional error term of e to the confidence intervals of our estimates for these marginals, and propagates to our learning guarantee. However, we do not currently know how the robust identity tests might behave with an approximate sampler. We thank the reviewer for mentioning this useful generalization, as oftentimes an MCMC sampler is used for DPPs. **Minor comments:** Thank you, we will follow these.

Reviewer #4 We will make the writing more accessible, add illustrations, and include synthetic experiments. **L4 in the algorithm:** In section 5.1, we briefly explain what the $\chi^2 - \ell_1$ test does. We will make it clearer. **Testing against parametric DPPs:** Indeed, our results motivate a study of additional structural assumptions that may bypass the lower bound. Currently, we do not know which sub-classes may enable this. **Summary Statistics:** Indeed, one hopes to exploit the parametric structure of DPPs to design good summary statistics for testing. We use the marginals in the learning part (based on the DPP structure), and the χ^2-ℓ_1 tester computes $Z^{(m)}$. Our information-theoretic lower bound also guarantees that there are no better statistics for ℓ_1 testing. **Uniformity testing intuition:** Intuitively, the dependence on ϵ comes from Central limit theorem: To test if a coin is unbiased or has bias ϵ with the sum statistic, at least $\Omega(1/\epsilon^2)$ samples are needed. For the intuition of the dependence on N, consider this problem: we observe samples from either the uniform distribution over $2N$, or over N items. By to the Birthday Paradox, with $o(\sqrt{N})$ samples, most likely no repetition of samples happens. But, without that, the two distributions are intuitively indistinguishable.
Negative Dependence: Indeed, our lower bound relies only on submodularity, and does not directly generalize to every class with the negatively dependence property. **Identifiability of K:** Indeed, during the generation of $\mathcal{M}$, we do not classify different kernel representations of a DPP. Reducing the repetitions reduces the size of $\mathcal{M}$.