We thank the reviewers for their thorough reading of our work. We will use their remarks to improve our draft.

Reviewer #1: ■ *The actual quantitative bounds for approximating a cost function using nonnegative features are difficult to parse and interpret.* We agree this is a bit dry. Our result in Prop. 3.1 gives a uniform control of the *ratio* between the kernel and its approximation. This can be compared to the classical result in [46] to control the *difference* between them. In the Sinkhorn setting, controlling the ratio is more important than controlling the difference (L.161), because it ensures that *the approximation retains the same positive sign*. We will discuss this further. ■ *For example, the bounds in Lemma 1 depend exponentially on dimension, what should we take from this?* When studying the Gaussian kernel in Lemma 1, both constants ψ and V that are upper bounds. While we can envision ways to obtain tighter, dimension free upper bounds using additional assumptions, we do indeed pay a price in dimension here when tackling the Euclidean distance with full generality. ■ *In the experiment of Figure 1, which is supposed to compare this method with nyquist, the setup is not clear enough.* Indeed, this must be clarified. These two normal distributions are in $\mathbb{R}^2$. One of them has mean $(1,1)^T$ and identity covariance matrix I_2. The other has 0 mean and covariance $0.1 \times I_2$. The cost function is the square Euclidean metric and the feature map is that presented in Lemma 1.

Reviewer #2: ■ *Thus, this method is not suitable for all cost function c.* Indeed, while we do stress the ability of our low-rank kernels to approximate known kernels induced by classical distances, some distances will prove elusive. For instance, if the support of both measures coincide, if the distance of interest is *not* Hilbertian, the resulting square kernel matrix K is *not* even positive definite, and certainly not approximated with low rank factors. Obtaining positive factors whose product can approximate K is therefore totally hopeless. This is why we also stress a *constructive* approach in our work, one that stresses the importance of considering families of kernels for which Sinkhorn has linear time complexity. ■ *r is not guaranteed to be small though its dependency on n is $O(\log n)$.* The fact that the dependency of r on n is $O(\log n)$ does not really affect the required number of random features as it depends logarithmically on the number of samples. However the constant ψ has a more important role. In the specific case of the Gaussian kernel, it is true that the constant ψ exhibited in the paper depends exponentially in the dimension but recall that this is an upper bound and it may be loose. Moreover the dependence in the dimension may be removed in some cases: for example if the data lies in a specific region of the space, then one may drop the dependence in the dimension in the bound by sampling the features on this low dimensional manifold. ■ *The empirical studies are somehow weak as the authors only compares different methods on calculating the Sinkhorn distance between two normal distributions(Exp1).* We used this setup for its simplicity, but we agree that another experiment would be welcome. We will prepare it for our next version. Moreover we will also add an experiment in a high-dimensional setting, to see how the proposed method scales with respect to the dimension in the specific case of the square Euclidean cost. ■ *As $k = exp(-c/\epsilon)$, k should be smaller than 1.* Because we flip things upside down and start directly with a kernel (not a cost), and the kernel itself is learned in an adversarial manner as the product of positive factors, the kernel can take arbitrarily large values. If they were recast (somewhat artificially) as cost functions, these values would indeed correspond to a negative cost. We also take this opportunity to correct a typo in Table 2: Image x / Noise z should be flipped both in lines and columns descriptions. ■ *What is performance of Sinkhorn algorithm with other direct rank r approximation of matrix K such as rank-r SVD?* As we mention in L.48-51 (and is discussed in [3]) using arbitrary low-rank factorizations for K fails if no proper care is taken to ensure that each factor has positive entries. This is not guaranteed for SVD, neither in theory nor have we observed it in practice. This is needed because the Sinkhorn iterations in Alg.1 use elementwise *divisions* of kernel products. We will clarify. Additionally, r-SVD would not allow a linear regime.

Reviewer #4: ■ *The section on generative adversarial networks needs more detail.* We will add more explanations on how to implement our method to train a W-GAN in the final version. Indeed here we embed the image space into the feature map space thanks to two operations. The first one consists in taking an image and embedding it into a latent space thanks to the mapping f_γ and the second one is an embedding of this latent space into the feature space thanks to the feature map φ_θ. Here the mapping considered is a parametrized version of the feature map defined in Lemma 1 obtained from the Gaussian kernel. ■ *The authors mention an alternative approach via batching the input and training a normal W-GAN but there is no comparison with this approach either quantitative or qualitative.* We will also add a discussion to compare with other W-GANs. Indeed our method has mainly two advantages compared to the other W-GANs proposed in the literature. First, the computation of the Sinkhorn divergence is linear with respect to the number of samples which allow to largely increase the batch size when training a W-GAN and obtain a better approximation of the true Sinkhorn divergence. Second, our approach is fully differentiable and therefore we can directly compute the gradient of the Sinhkorn divergence with respect the parameters of the network. In [48] the authors do not differentiate through the Wasserstein cost to train their network. In [27] the authors do differentiate through the iterations of the Sinkhorn algorithm but this strategy require to keep track of the computation involved in the Sinkhorn algorithm and can be applied only for large regularizations as the number of iterations cannot be too large. ■ *It would also be great to see how the approach compares to Sinkhorn/Nystrom on different manifolds and metrics.* We will add another experiment to compare our proposed approach with the Nystrom method on different manifolds in the final version.