The Quantum Trellis: A classical algorithm for sampling the parton shower with interference effects

Sebastian Macaluso
New York University
seb.macaluso@nyu.edu

Kyle Cranmer
New York University
kyle.cranmer@nyu.edu

Abstract

Simulations of high-energy particle collisions, such as those used at the Large Hadron Collider, are based on quantum field theory; however, many approximations are made in practice. For example, the simulation of the parton shower, which gives rise to objects called 'jets', is based on a semi-classical approximation that neglects various interference effects. While there is a desire to incorporate interference effects, new computational techniques are needed to cope with the exponential growth in complexity associated to quantum processes. We present a classical algorithm called the *quantum trellis* to efficiently compute the un-normalized probability density over N-body phase space including all interference effects, and we pair this with an MCMC-based sampling strategy. This provides a potential path forward for classical computers and a strong baseline for approaches based on quantum computing.

1 Introduction

The high-energy particle physics community relies on high-fidelity predictions of particle collisions. These predictions are based on quantum field theory, but in practice many approximations are made. While tools like `MadGraph` [Alwall et al., 2011] model the *hard* collision and include interference effects between different Feynman diagrams, the parton shower implemented in tools like `Pythia` [Sjostrand et al., 2006] are based on a semi-classical approximation that neglects various interference effects. In particular, the classical treatment of the parton shower admits an efficient sampling algorithm where the shower evolves sequentially and is described by an autoregressive probabilistic model (a Markov process). There is a desire to improve upon the classical treatment and explicitly incorporate interference effects [Nagy and Soper, 2008, 2014]; however, new computational techniques are needed to cope with the exponential growth in complexity associated to quantum processes [Provasoli et al., 2019, Bauer et al., 2021a,b].

Contributions of this paper We present a classical data structure (the *quantum trellis*) and dynamic programming algorithm to efficiently compute the un-normalized probability density over N-body phase space including all interference effects and pair this with a MCMC-based sampling strategy. This provides a potential path forward for sampling the parton shower including interference effects with classical computers and a strong baseline for approaches based on quantum computing.

2 The classical trellis

The hierarchical trellis described in Macaluso et al. [2021] is a data structure that can be paired with a dynamic programming algorithm to efficiently search or sum over the enormous space $\mathcal{H}$ of hierarchical clusterings of N objects. It generalizes a previously developed algorithm described in Greenberg et al. [2018] for flat clustering. The hierarchical generalization was motivated by the study of 'jets' at the LHC [Cranmer et al., 2021a]. In that context, the N objects to be clustered correspond to final state particles observed in the large particle detectors like ATLAS and CMS. Each of those particles have energy and momentum as features $x_i \in \mathbb{R}^4$. The hierarchical clusterings $\mathtt{H} \in \mathcal{H}$

Fourth Workshop on Machine Learning and the Physical Sciences (NeurIPS 2021).

correspond to one of the possible latent showering histories that could give rise to $X = \{x_i\}_{i=1}^N$ via the parton shower (see Figure 1). Many tasks in jet physics involve either searching for the most likely showering history $\hat{\mathtt{H}}(X)$ or calculating the marginal likelihood by summing over all possible showering histories. The computational difficulty lies in the fact that there are $(2N-3)!!$ possible hierarchical clusterings and an additional 2^{N-1} permutations of the left/right children at each binary splitting[1]. Therefore, brute-force search or sum is not feasible for even $N \approx 10$.

Macaluso et al. [2021] considered a so-called "energy-based" probabilistic model for hierarchical clustering, though "energy" is not to be taken literally in the physics context[2]. In particular, the hierarchical trellis and dynamic programming algorithms were specialized to situations where the "energy" $\phi(X|\mathtt{H})$ is based on measuring the compatibility of each pair of sibling nodes, described by a "potential function" $\psi(X_L, X_R)$. The posterior probability $P(\mathtt{H}|X)$ of $\mathtt{H}$ for the dataset X is equal to the unnormalized potential of $\mathtt{H}$ normalized by the partition function, $Z(X)$:

$$P(\mathtt{H}|X) = \frac{\phi(X|\mathtt{H})}{Z(X)} \quad \text{with} \quad \phi(X|\mathtt{H}) = \prod_{X_L, X_R \in \text{siblings}(\mathtt{H})} \psi(X_L, X_R) \tag{1}$$

where the partition function $Z(X)$ is given by:

$$Z(X) = \sum_{\mathtt{H} \in \mathcal{H}(X)} \phi(X|\mathtt{H}). \tag{2}$$

and $\mathcal{H}(X)$ gives all binary hierarchical clusterings of the elements X.

The resulting algorithm's computational complexity is $\mathcal{O}(3^N)$. While still exponential, it is a super-exponential improvement over a naive iteration over the hierarchies $\mathtt{H} \in \mathcal{H}$. This makes it feasible in regimes where enumerating all possible trees would be infeasible, and is to our knowledge the fastest exact partition function result, making practical exact inference for datasets on the order of 20 points (~3×10^9 operations vs ~10^{22} trees).

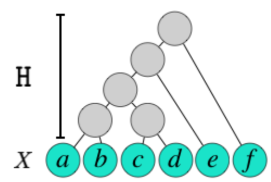

Figure 1: Schematic representation of a hierarchical clustering $\mathtt{H}$ for the dataset X.

3 The shower model

Classical Ginkgo The autoregressive form of the potential function $\phi(X|\mathtt{H})$ in Eq. 1 as a product over splittings where each splitting contributes $\psi(X_L, X_R)$ is consistent with the classical description of the parton shower. However, in practice state-of-the-art parton showers do not expose the splitting likelihoods in a convenient way. Thus, to aid in machine learning research for jet physics, a python package for a simplified generative model of a parton shower, called `Ginkgo`, was introduced in Cranmer et al. [2019, 2021b]. `Ginkgo` exposes the probability model for each splitting and has a tractable joint likelihood $p(x|\mathtt{H})$. Each edge in the tree corresponds to a particle with an energy-momentum vector $x = (E \in \mathbb{R}^+, \vec{p} \in \mathbb{R}^3)$ and squared mass $t(x) = E^2 - |\vec{p}|^2$. A parent's energy-momentum vector is obtained from adding its children, i.e., $x_P = x_L + x_R$. For each splitting with $(\sqrt{t_P}, \sqrt{t_L}, \sqrt{t_R})$ for the masses of the parent, left-, and right-child the potential function is given by $\psi(X_L, X_R) = f(t_L|t_P, \lambda) f(t_R|t_P, \lambda)$, where

$$f(t|t_P, \lambda) = \frac{1}{1 - e^{-\lambda}} \frac{\lambda}{t_P} e^{-\lambda \frac{t}{t_P}} \ . \tag{3}$$

Note, the first term in $f(t|t_P, \lambda, \beta)$ is a normalization factor associated to the constraint that $t < t_P$.

While it is a simplification, it captures essential ingredients of parton shower generators. Within the analogy between jets and natural language processing (NLP), `Ginkgo` can be thought of as a generative language model that produces $(X, \mathtt{H})$ pairs where X is the text, H is the ground-truth parse trees, and $p(X|\mathtt{H})$ is known.

[1](see Callan [2009], Dale and Moon [1993] for more details and proof).

[2]This is refered to as an energy-based model since often it is the case that $\psi(\cdot, \cdot)$ has the form of an unnormalized Gibbs distribution, as $\psi(X_L, X_R) = \exp(-\beta E(X_L, X_R))$, where β is the inverse temperature and $E(\cdot, \cdot)$ is the energy.

The analogous quantum mechanical amplitude In order to study the quantum analogue of the `Ginkgo` parton shower model, we must have a quantum mechanical amplitude $\mathcal{A}(X|\mathtt{H})$ for a given hierarchy $\mathtt{H}$. We base our amplitude on the classical generative model implemented in `Ginkgo`. In keeping with what one expects from Feynman rules, we define the amplitude of a hierarchy $\mathcal{A}(X|\mathtt{H})$ as the product of the amplitudes $\mathcal{A}(x_L, x_R)$ for all the $1 \to 2$ splittings, which includes a complex phase that depends on the invariant mass of the parent. Specifically, we choose

$$\mathcal{A}(x_L, x_R; \lambda, \beta) = e^{-i\beta t_P} \sqrt{f(t_L|t_P, \lambda)} \sqrt{f(t_R|t_P, \lambda)} \tag{4}$$

where the first term is a t_P-dependent complex phase with hyperparameter β and the square-roots are introduced to maintain consistency with the splitting likelihoods used in the classical Ginkgo model in Eq.3. Note that in the case where $\beta = 0$ each term is real, but there is still constructive interference when calculating $|\mathcal{A}(X|\mathtt{H})|^2$. Therefore $\beta = 0$ does not correspond to the classical case. We would like to end this section by emphasizing that the quantum mechanical amplitude defined by Eq. 4 is not meant to be physically justified, but to have the right form for exploring efficient algorithms for parton showers with quantum interference.

4 Quantum Hierarchical trellis

The basic idea for the *quantum trellis* is simple: replace the real-valued potential function $\psi(x_L, x_R)$ with the complex-valued amplitude $\mathcal{A}(x_L, x_R)$. Each path through the trellis will correspond to a particular hierarchy $\mathtt{H}$, and one can accumulate the terms to compute $\mathcal{A}(X|\mathtt{H}) = \prod_{x_L, x_R \in \text{siblings}(\mathtt{H})} \mathcal{A}(x_L, x_R)$ as before. Then we can use the same hierarchical trellis data structure and dynamic programming algorithm to efficiently compute the total amplitude for all possible $(2N-3)!!$ showering histories $\mathcal{A}(X) = \sum_{\mathtt{H} \in \mathcal{H}(X)} \mathcal{A}(X|\mathtt{H})$.[3]

Using the Born rule, the un-normalized probability density $\tilde{p}(X)$ is given by the square of the magnitude of the total amplitude

$$\tilde{p}(X) = \left| \sum_{\mathtt{H} \in \mathcal{H}(X)} \mathcal{A}(X|\mathtt{H}) \right|^2 . \tag{5}$$

The distribution $\tilde{p}(X)$ includes all constructive and destructive interference among the different hierarchies. In principle, this includes all $((2N-3)!!)^2$ cross-terms as schematically illustrated in Figure 2; however, one need not explicitly construct cross terms $\mathcal{A}_i \mathcal{A}_j^*$ if one simply performs the sum before squaring.

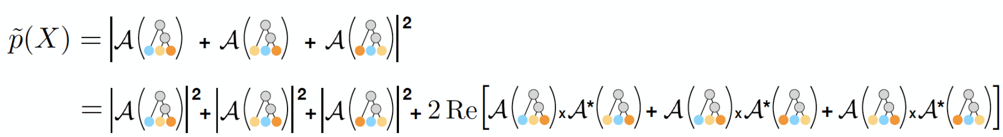

Figure 2: Schematic representation of all the terms of the marginal likelihood for a dataset of three elements, showing the interference terms.

To the best of our knowledge, this is the first time the marginal amplitude can be exactly obtained over datasets of $\mathcal{O}(10)$ elements. For example, with $N = 8$ there are 135,135 possible hierarchies and over 18 billion cross terms.

5 Sampling including interference effects

The final goal is to be able to generate events according to the distribution including all interference effects, e.g. we want to sample $X \sim p(X)$. The normalized distribution is given by $p(X) = \tilde{p}(X) / \int dX \tilde{p}(X)$, but the normalizing factor in the denominator corresponds to a challenging integral over phase space. In this work, we employ Markov Chain Monte Carlo (MCMC) techniques, which only need the target distribution defined up to a multiplicative constant. To perform MCMC sampling we use the emcee library from Foreman-Mackey et al. [2013] and fix the number of final state particles N. Correctly generating the distribution over N is a challenging problem as it involves the phase space integrals mentioned above, and is left for future work.

[3] The Cluster Trellis code can be accessed at https://github.com/SebastianMacaluso/ClusterTrellis

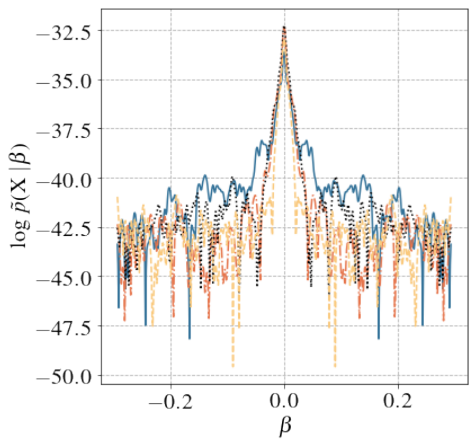

Figure 3: Log of the un-normalized likelihood $\tilde{p}(X|\beta)$ vs the complex phase β for four independent X with 8 leaves sampled from the classical Ginkgo model. We see that all datasets have a similar behavior, with a peak at zero where there is only constructive interference.

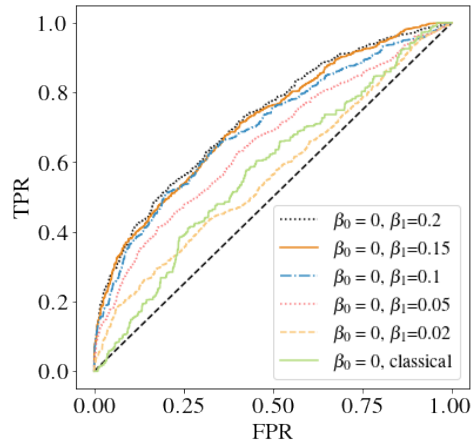

Figure 4: ROC curves between pairs of datasets (with 8 leaves) sampled with MCMC for different complex phase β. We also show the ROC curve between the $\beta = 0$ model and classical Ginkgo (solid green).

6 Experiments: Jet Physics

We generated datasets using the MCMC technique described above as well as the classical Ginkgo generative model. These datasets of 44000 samples are obtained by running MCMC for 550 steps (with additional 50 burn-in steps) and 80 walkers. It took about 30 hours to generate each of the 44000 samples on an Intel Xeon Platinum 8268 24C 2.9GHz Processor.

Results In Figure 3, we show the dependence of $\tilde{p}(X|\beta)$ on the hyperparameter β for four independent X sampled from the classical `Ginkgo` model. We see that each has a similar behavior with the un-normalized likelihood peaking at $\beta = 0$ where we only have constructive interference.

Next, we characterize the effect of interference via the ROC AUC between datasets $X \sim p(X|\beta = 0)$ and $X \sim p(X|\beta = \beta_1)$, for different values of β_1. With the trellis algorithm, we are able to directly calculate the discrimination power of the optimal classifier without training a classifier [4]. We show in Figure 4 the ROC curve for $\tilde{p}(X)$, between pairs of datasets with 8 leaves for different values of β_1. We also show the ROC curve between the $\beta = 0$ model and classical `Ginkgo`. We can see that MCMC together with the quantum trellis allow to generate samples that are different from the ones generated with classical `Ginkgo`.

7 Conclusion

We developed the quantum trellis data structure and dynamic programming algorithm that allows us to efficiently calculate the probability using the Born rule, as well as a MCMC technique to sample from $\tilde{p}(X)$ including all interference effects. The resulting approach provides a strong classical baseline for potential quantum algorithms that might be used to sample from a parton shower including interference effects.

We end by noting that Provasoli et al. [2019], Bauer et al. [2021a,b]) considered quantum algorithms for simulating a similar system: the binary random walk of a single particle taking N steps to the left or right. In that case, the state space looks like a binary tree with 2^N leaves, but each realization is a single path. They developed efficient algorithms to sample the quantum process where multiple paths interfere. Our case is exponentially harder as each realization is not a single path but itself a binary tree, thus the state space corresponds to the $(2N-3)!!$ showering histories for N particles.

[4]The optimal classifier is based on the Neyman–Pearson lemma and defined by the likelihood ratio as the most powerful variable or test statistic (for a proof and a particle physics application see J. Stuart and Arnold [1994], Cranmer and Plehn [2007]).

8 Broader Impact

Hierarchical clustering is often used to discover meaningful structures, such as phylogenetic trees of organisms Kraskov et al. [2005], taxonomies of concepts Cimiano and Staab [2005], subtypes of cancer Sørlie et al. [2001], and jets in particle physics Cacciari et al. [2008]. Previous work with the hierarchical trellis is particularly relevant in situations where one would like to consider many such clusterings weighted by a domain-motivated energy function. Providing computationally efficient means to consider all such clusterings enables the treatment of uncertainty and other probabilistic concepts, which can aid in the responsible use of such clusterings for down-stream tasks.

The current work extends the probabilistic modelling of individual hierarchies to a quantum setting where interference effects are important. As such, the domain of application is much smaller. The original motivation is that these developments will lead to improved simulation of jets at the Large Hadron Collider.

In addition to the direct target application of jet physics at the LHC, the algorithm described here provides a strong baseline for quantum algorithms that try to solve the same sampling problem. As such, they inform more broadly the development of quantum algorithms and quantum computing in a broader setting, similar to the sampling problem studied in Arute et al. [2019].

Acknowledgements

Kyle Cranmer and Sebastian Macaluso are supported by the National Science Foundation under the awards ACI-1450310 and OAC-1836650 and by the Moore-Sloan data science environment at NYU. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

References

J. Alwall, M. Herquet, F. Maltoni, O. Mattelaer, and T. Stelzer. MadGraph 5 : Going Beyond. *JHEP*, 06:128, 2011. doi: 10.1007/JHEP06(2011)128.

F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. Brandao, D. A. Buell, et al. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, 2019.

C. W. Bauer, W. A. de Jong, B. Nachman, and D. Provasoli. Quantum Algorithm for High Energy Physics Simulations. *Phys. Rev. Lett.*, 126(6):062001, 2021a. doi: 10.1103/PhysRevLett.126.062001.

C. W. Bauer, M. Freytsis, and B. Nachman. Simulating collider physics on quantum computers using effective field theories. 2 2021b.

M. Cacciari, G. P. Salam, and G. Soyez. The anti-k_t jet clustering algorithm. *JHEP*, 04:063, 2008. doi: 10.1088/1126-6708/2008/04/063.

D. Callan. A combinatorial survey of identities for the double factorial, 2009.

P. Cimiano and S. Staab. Learning concept hierarchies from text with a guided agglomerative clustering algorithm. In *Proceedings of the ICML 2005 Workshop on Learning and Extending Lexical Ontologies with Machine Learning Methods*, 2005.

K. Cranmer and T. Plehn. Maximum significance at the lhc and higgs decays to muons. *The European Physical Journal C*, 51(2):415–420, Jun 2007. ISSN 1434-6052. doi: 10.1140/epjc/s10052-007-0309-4. URL `http://dx.doi.org/10.1140/epjc/s10052-007-0309-4`.

K. Cranmer, S. Macaluso, and D. Pappadopulo. Simplified Generative Model for Jets Package, 2019. https://github.com/SebastianMacaluso/ginkgo.

K. Cranmer, M. Drnevich, S. Macaluso, and D. Pappadopulo. Reframing Jet Physics with New Computational Methods. *EPJ Web Conf.*, 251:03059, 2021a. doi: 10.1051/epjconf/202125103059.

K. Cranmer, M. Drnevich, S. Macaluso, and D. Pappadopulo. Reframing jet physics with new computational methods. *EPJ Web of Conferences*, 251:03059, 2021b. ISSN 2100-014X. doi: 10.1051/epjconf/202125103059. URL `http://dx.doi.org/10.1051/epjconf/202125103059`.

E. Dale and J. Moon. The permuted analogues of three Catalan sets, 1993.

D. Foreman-Mackey, D. W. Hogg, D. Lang, and J. Goodman. emcee: The mcmc hammer. *Publications of the Astronomical Society of the Pacific*, 125(925):306–312, Mar 2013. ISSN 1538-3873. doi: 10.1086/670067. URL `http://dx.doi.org/10.1086/670067`.

C. Greenberg, N. Monath, A. Kobren, P. Flaherty, A. McGregor, and A. McCallum. Compact representation of uncertainty in clustering. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 8630–8640. Curran Associates, Inc., 2018. http://papers.nips.cc/paper/8081-compact-representation-of-uncertainty-in-clustering.pdf.

A. O. J. Stuart and S. Arnold. Kendall's advanced theory of statistics. In *Vol 2A (6th Ed.) (Oxford University Press, New York*, 1994.

A. Kraskov, H. Stögbauer, R. G. Andrzejak, and P. Grassberger. Hierarchical clustering using mutual information. *EPL (Europhysics Letters)*, 70(2):278, 2005.

S. Macaluso, C. Greenberg, N. Monath, J. Ah Lee, P. Flaherty, K. Cranmer, A. McGregor, and A. McCallum. Cluster trellis: Data structures & algorithms for exact inference in hierarchical clustering. In A. Banerjee and K. Fukumizu, editors, *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pages 2467–2475. PMLR, 13–15 Apr 2021. URL `https://proceedings.mlr.press/v130/macaluso21a.html`.

Z. Nagy and D. E. Soper. Parton showers with quantum interference: Leading color, with spin. *JHEP*, 07:025, 2008. doi: 10.1088/1126-6708/2008/07/025.

Z. Nagy and D. E. Soper. A parton shower based on factorization of the quantum density matrix. *JHEP*, 06:097, 2014. doi: 10.1007/JHEP06(2014)097.

D. Provasoli, B. Nachman, W. A. de Jong, and C. W. Bauer. A Quantum Algorithm to Efficiently Sample from Interfering Binary Trees. 1 2019.

T. Sjostrand, S. Mrenna, and P. Z. Skands. PYTHIA 6.4 Physics and Manual. *JHEP*, 05:026, 2006. doi: 10.1088/1126-6708/2006/05/026.

T. Sørlie, C. M. Perou, R. Tibshirani, T. Aas, S. Geisler, H. Johnsen, T. Hastie, M. B. Eisen, M. Van De Rijn, S. S. Jeffrey, et al. Gene expression patterns of breast carcinomas distinguish tumor subclasses with clinical implications. *Proceedings of the National Academy of Sciences*, 98(19): 10869–10874, 2001.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] We discuss simplifications of the shower model and limitations of sampling to a fixed number of particles. Also, while the current algorithm is efficient, it is still exponential in N.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] Potential negative societal impacts were considered, and we believe this work does not present any issues in this regard.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] We discussed the assumptions on the form of the potential function / amplitude either explicitly or as described in referenced works.
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] The code repository associated with this paper is referenced.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [N/A] No learning algorithms were used in this work.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A] No learning algorithms were used. The ROC curve has some variability due to the MCMCM samples. These were not yet included due to limited computation time.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [N/A] Licenses are mentioned in the links associated with individual code packages.
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A] No new assets (excluding the code used to reproduced the experiments) were produced in this work.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] No personal information is included in the assets utilized in this paper.
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]