| | RMSE | | | | | | Time (seconds) | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | **ours** (20%) | **ours** (80%) | **MICE** (20%) | **MICE** (80%) | **MC** (20%) | **MC** (80%) | **ours** (learning) | **ours** (prediction) | **MICE** | **MC** |
| Abalone($1e0$) | 2.72$\pm$.04 | 2.99$\pm$.02 | 2.71$\pm$.07 | 4.50$\pm$.09 | 2.80$\pm$.04 | 3.39$\pm$.07 | 82 | 20 | 43 | 117 |
| Delta ($1e-4$) | 1.84$\pm$.02 | 2.61$\pm$.05 | 1.88$\pm$.02 | 3.27$\pm$.06 | 1.91$\pm$.02 | 2.80$\pm$.04 | 53 | 24 | 27 | 126 |
| Insurance($1e3$) | 5.84$\pm$.24 | 10.1$\pm$.43 | 6.24$\pm$.38 | 13.0$\pm$.77 | 6.14$\pm$.28 | 10.1$\pm$.49 | 40 | 13 | 11 | 20 |
| Elevators($1e-2$) | 0.46$\pm$.01 | 0.64$\pm$.01 | 0.44$\pm$.01 | 1.15$\pm$.04 | 0.55$\pm$.01 | 0.83$\pm$.01 | 2105 | 31 | 364 | 994 |

We thank the reviewers for their valuable feedback. We appreciate they recognize that the paper is "*well-written*" and "*clear*" (R#2, R#3, R#4, R#5), whose technical contribution "*quality is solid*" (R#2), "*very good*" (R#1, R#5) and "*non-trivial*" (R#3) while it considers "*an important problem in ML*" (R#3, R#4) which can "*be of interest to many people at NeurIPS*" (R#3). We hope to address all questions and concerns raised in the following.

**[Reviewer #1]** **1. Limited impact.** We disagree with the reviewer. As also noted by R3 and R4, computing the expected predictions of a model lies *at the core of ML and statistics*. Among the plethora of ML problems that would benefit from our algorithm, there are: missing value imputation, feature selection, several formulations of fairness as well as computing integral probability metrics, i.e., a fundamental way to assess the distance between distributions (e.g., see the popular Wasserstein distance). In this paper, we tackled just the first one in the list to show the effectiveness of our algorithm. We are actively working on applying it to the other application scenarios. **2. Toy models.** the structural properties we require for our circuits *do not compromise expressiveness*: PSDDs are SOTA density estimators that are comparable to MADEs and VAEs on many benchmarks (compare the results in [1] w.r.t. those in [2]) and LCs are able to achieve the same accuracy of much more complex neural networks (e.g., Resnets cfr. [3]).

**[Reviewer #2]** **1. Results easily follow from literature.** Our technical contribution goes beyond the results known in the literature of circuits. Classic sum/max problems only require simpler structural properties and they focus on one circuit at a time. E.g., sums (marginals) require only decomposability and smoothness, with the addition of determinism for max problems (MAP). Here, for expectations, we need to deal with a pair of circuits and we require them to be both structured decomposable and to share the same vtree. We agree that computations are simple, i.e., elegant, *once the aforementioned requirements have been elicited*. Eliciting them, however, is definitely non-trivial and has not been explored in the literature so far for expectations. Indeed, our work has been made possible only very recently, after discriminative circuits satisfying such structural properties have been introduced in [2]. **2. Simple datasets.** Statistics are reported in the Appendix. Note that our contribution is more theoretical than empirical. As such, our experiments are meant to showcase the (theoretically expected) effectiveness of our algorithms when a reasonably accurate generative model is available, across different real world datasets. Our circuits are expressive enough to model larger datasets (see our answer to R#1.2) and learning them would scale: in many cases it is easier to learn a LC than a neural net (e.g., see [3]). **3. Approximate inference alternatives.** Whenever we are able to compute expectations exactly for regression (Thm 1), we might want to consider approximations only to speed computations. This is however not necessary in practice, as our algorithm is very efficient due to caching (see next point). For classification, we resort to approximations but, unfortunately, we cannot provide anytime guarantees. We will discuss and cite related works on anytime approximations as it is a sensible venue to explore. **4. Run times.** We report in the top table the RMSE and the avg. time to predict one test sample for regression with 20% and 80% missing values (we will report all results in the paper) and compare to Monte Carlo (MC) estimates via 200 samples drawn from the PSDD. Our method is not only faster but more accurate than MC (and MICE). Note that the time to learn the regression circuit is easily amortized after the prediction of a few samples. **5. Code and figures.** We will make the figures and code more accessible.

**[Reviewer #3]** **Related works.** We will add a detailed discussion of previous approaches to computing moments, such as Monte Carlo methods (along with experiments; see response R#2.4) and missing value imputation techniques.

**[Reviewer #4]** **1. Proofs.** We will provide more detailed proofs for Thms 2 & 3. Specifically, we will show in detail how we can reduced our case to those whose complexity has been previously derived. **2. Extension.** The work in [4] avoids computing expectations by distilling a (simple) generative model from a (simple) discriminative model. We take another path, which is not a trivial derivation. See also our answer to R#2.1. **3. Negative results.** For regression (Thm 2), the needed structural constraints do not hinder expressiveness. See our answer to R#1.2. For classification (Thm 3), we need to resort to approximations (which are still more effective than competitors for missing values). Note that Thm 3 does not state that there cannot exist a circuit pair with additional structural assumptions enabling exact computations.

**[Reviewer #5]** **1. Wrong audience.** Our method can be impactful to many ML scenarios (see our answer to R#1.1). As R#3 and R#4 recognize, NeurIPS is a sensible venue. **2. Finite data.** We exploit a generative model as a proxy to the true data distribution. Indeed, we learn it from data, and the better density estimator it is, the more accurate the expected predictions will be. We will discuss this in Section 3 along with how to deal with continuous data. **3. Baselines.** We will add the comparison with MC estimates over samples from the same PSDD (see our answer to R#2.4).

**[References]** **[1]** Liang et al. "Learning the Structure of Probabilistic Sentential Decision Diagrams" UAI 2017 **[2]** Peharz et al. "Random Sum-Product Networks: A Simple and Effective Approach to Probabilistic Deep Learning" UAI 2019 **[3]** Liang et al. "Learning logistic circuits" AAAI 2019 **[4]** Khosravi et al. "What to expect of classifiers?" IJCAI 2019