1. Motivation

- PPLs are extremely powerful and flexible, but this makes inference hard
  - PPLs have a scaling problem: focus primarily on small programs

  **Our goal:** Focus on discrete programs, make a high-performance exact inference algorithm for this specialized setting

  - Discreteness is very common, many programs have discrete parts (text, graphs, computer networks, ...)
  - Discreteness is challenging for many methods
    - Many methods rely on differentiability
    - Low-probability observations
  - Exact inference preferable to approximate
    - Does not propagate errors
    - Suitable for high-consequence decisionmaking

2. Method

- Key idea: factorize the inference computation (see Figure 1a)

\[
\begin{align*}
0.1 
\times (0.2 
\times 0.4 
+ 0.1 
\times 0.8 
\times 0.5 
+ 0.9 
\times 0.3 
\times 0.4 
+ 0.9 
\times 0.7 
\times 0.5 
) 
\end{align*}
\]

Versus...

\[
\begin{align*}
0.1 
\times (0.2 
\times 0.4 
+ 0.8 
\times 0.5 
) 
+ 0.9 
\times (0.3 
\times 0.4 
+ 0.7 
\times 0.5 
) 
\end{align*}
\]

- Finding and exploiting these factorization opportunities can be hard!
- We do it with binary decision diagrams (BDDs)

3. Experiments

- Show Dice can perform exact inference on extremely large programs
  - For instance, a 1.9 megabyte program with over 100k random variables

- Compared Dice against Psi and Ace (specialized Bayesian network solver)

- Three main experiments:
  1. Common Baselines
  2. Single-marginal Bayesian network inference
  3. All-marginal Bayesian network inference

4. Conclusion

- Github: https://github.com/SHoltzen/dice
- Webpage: http://dicelang.cs.ucla.edu/

5. Acknowledgments

This work is partially supported by NSF grants #IIS-1943641, #IIS-1956441, #CCF-1837129, DARPA grant #N66001-17-2-4032, a Sloan Fellowship, and gifts by Intel and Facebook research. The authors would like to thank Jon Aytaç and Philip Johnson-Freyd for feedback on paper drafts.