

Bit Blasting Probabilistic Programs

Poorva Garg, Steven Holtzen, Guy Van den Broeck, Todd Millstein

UCLA



FPTalks - July 11 2024

What are Probabilistic Programs?

Programs that represent
probability distributions:

```
a ~ flip(0.7)
b ~ if a
    then normal(0, 1)
    else normal(2, 1)
return b
```

Primary analysis task is
probabilistic inference:

$$\begin{aligned} pr(b) &= \sum_{a_i} pr(a = a_i) pr(b|a = a_i) \\ &= \frac{7}{10} pr(b|a = 1) + \frac{3}{10} pr(b|a = 0) \\ &= \frac{7}{10} e^{-\frac{1}{2}b^2} + \frac{3}{10} e^{-\frac{1}{2}(2-b)^2} \end{aligned}$$

What are Probabilistic Programs?

Programs that represent
probability distributions:

```
a ~ flip(0.7)
b ~ if a
    then normal(0, 1)
    else normal(2, 1)
return b
```

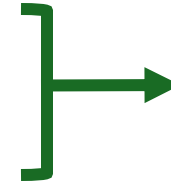
Primary analysis task is
probabilistic inference:

$$\begin{aligned} pr(b) &= \sum_{a_i} pr(a = a_i) pr(b|a = a_i) \\ &= \frac{7}{10} pr(b|a = 1) + \frac{3}{10} pr(b|a = 0) \\ &= \frac{7}{10} e^{-\frac{1}{2}b^2} + \frac{3}{10} e^{-\frac{1}{2}(2-b)^2} \end{aligned}$$

discrete + **continuous** = **hybrid** probabilistic program

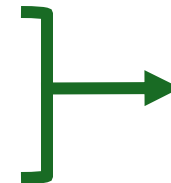
Hybrid is Not Well Supported

Hamiltonian Monte Carlo



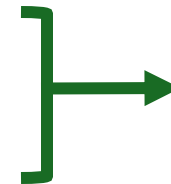
Limited support
for discreteness

Sequential Monte Carlo



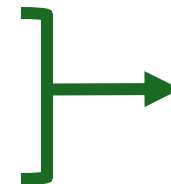
Scalability and
accuracy issues

Algebraic Evaluation



Closed form does
not always exist

Knowledge Compilation



No support
for continuous

Bit Blasting a Continuous Random Variable

Infinite binary representation in $[0,1)$:

$$X \sim 0. b_1 b_2 b_3 \dots$$

- ✓ all random variables are discrete
- ✓ representation is exact
- ✓ exposes useful structure (e.g., arithmetic)
- ✗ infinite number of random bits

Bit Blasting a Continuous Random Variable

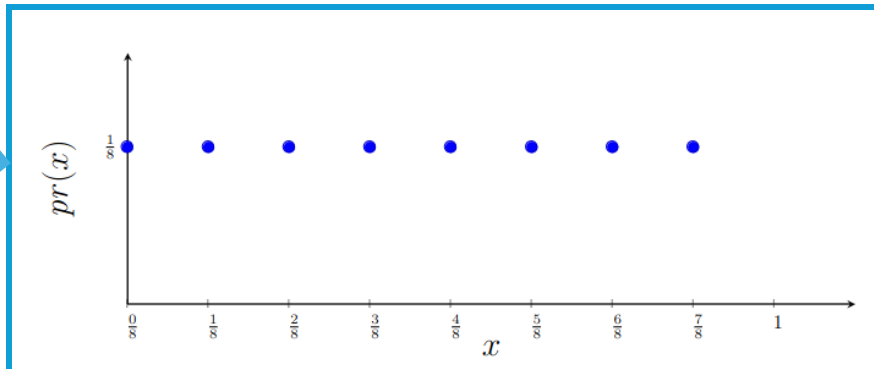
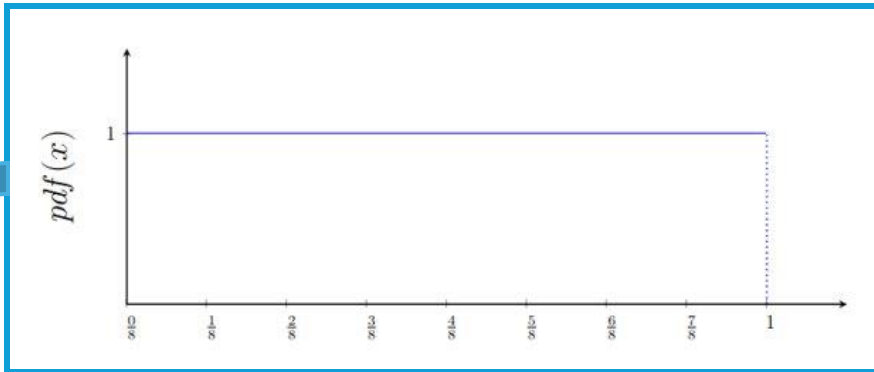
Finite binary representation in $[0,1)$:

$$X \sim 0. b_1 b_2 b_3 \dots b_k$$

- ✓ all random variables are discrete
- ✓ representation is exact **up to k bits**
- ✓ exposes useful structure (e.g., arithmetic)
- ? **does the distribution over k bits have a program using a few independent coin flips?**

Bit Blasting the Uniform

$$X \sim 0. b_1 b_2 b_3$$



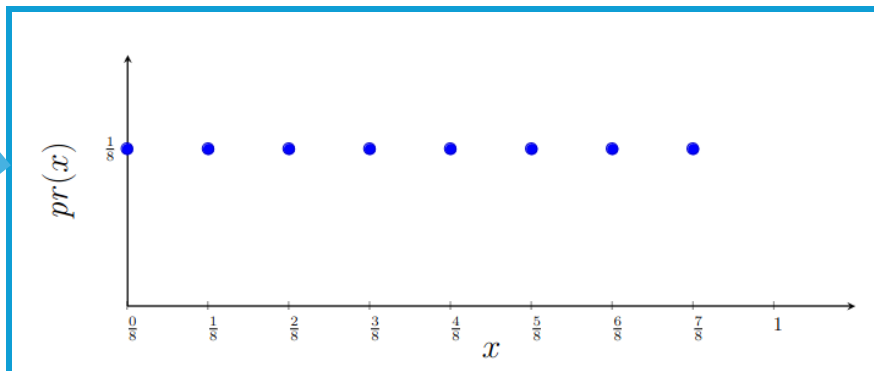
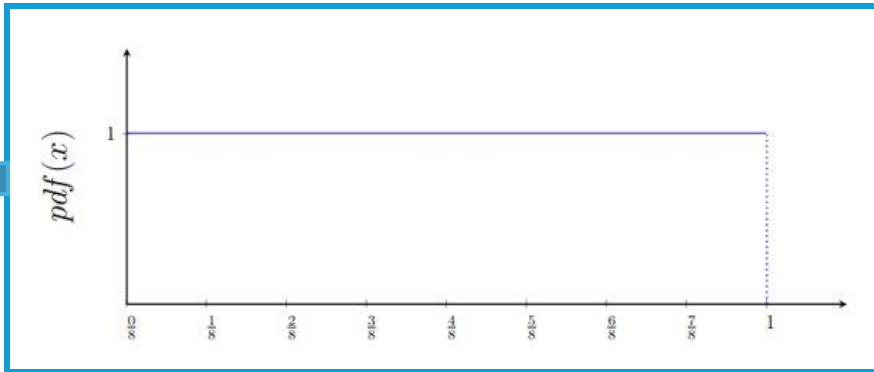
...

*represent bits using
a probabilistic program
of coin flips*

...

Bit Blasting the Uniform

$$X \sim 0.b_1b_2b_3$$



Naive discretization

```
if flip(1/8) [0, 0, 0]
elif flip(1/7) [0, 0, 1]
elif flip(1/6) [0, 1, 0]
elif flip(1/5) [0, 1, 1]
elif flip(1/4) [1, 0, 0]
elif flip(1/3) [1, 0, 1]
elif flip(1/2) [1, 1, 0]
else [1, 1, 1] end
```

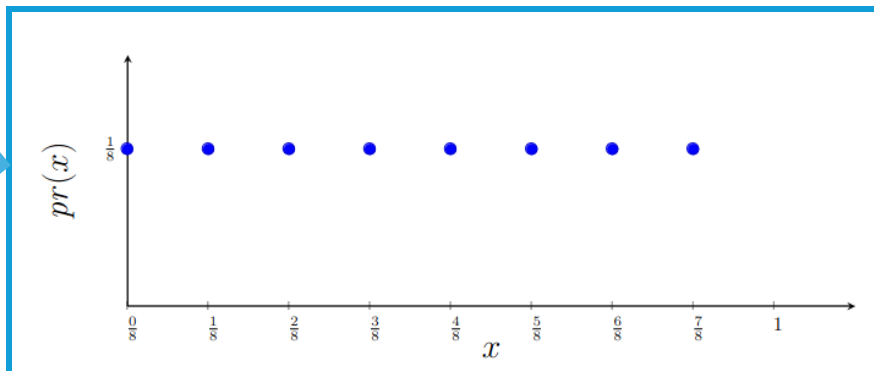
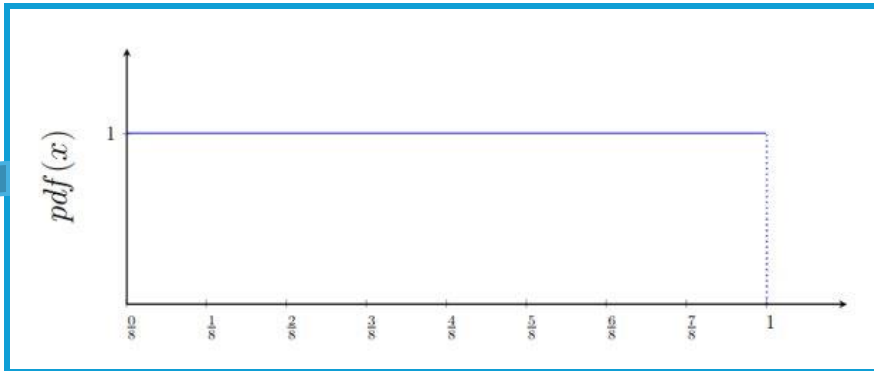
How many coin flips?

- 3 bits: 7 flips
- 32 bits: 4,294,967,295 flips
- k bits: $2^k - 1$ flips **×**

see GuBPI, AQUA, etc.

Bit Blasting the Uniform

$$X \sim 0.b_1b_2b_3$$



Bit Blast

`a = flip(0.5)`

`b = flip(0.5)`

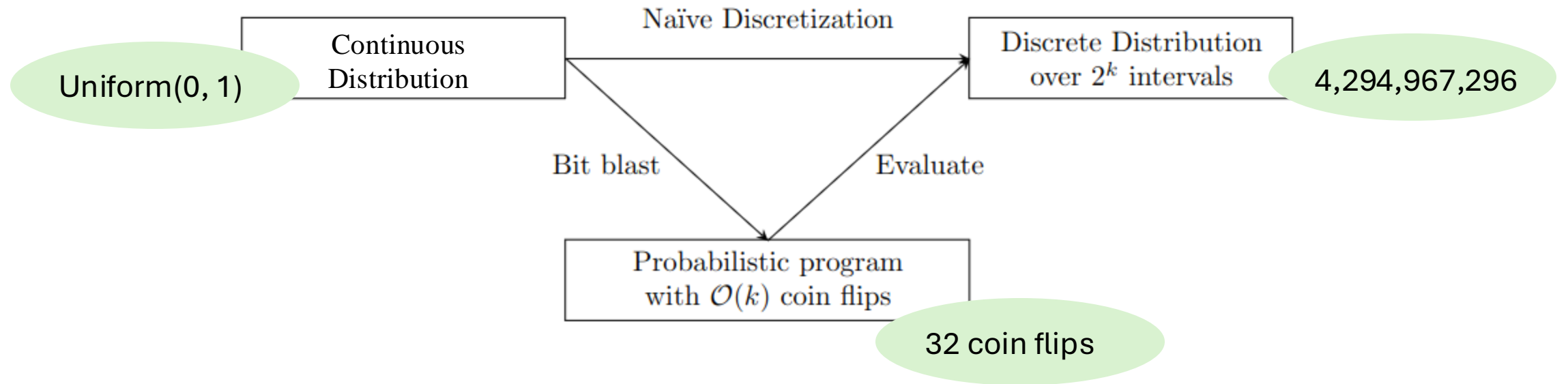
`c = flip(0.5)`

`[a, b, c]`

How many coin flips?

- 3 bits: 3 flips
- 32 bits: 32 flips
- k bits: k flips ✓

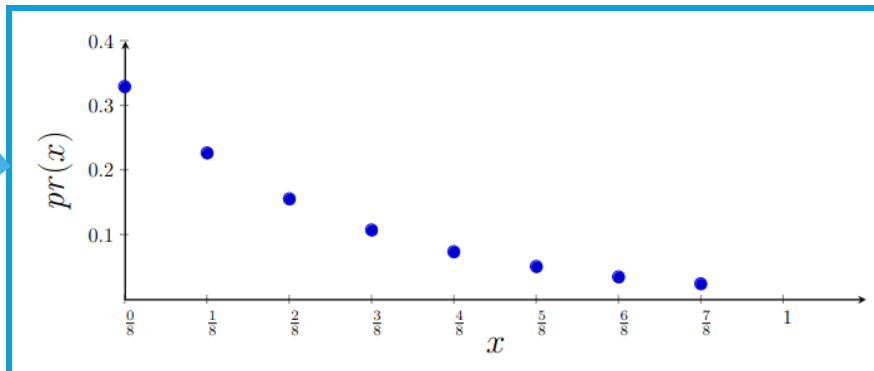
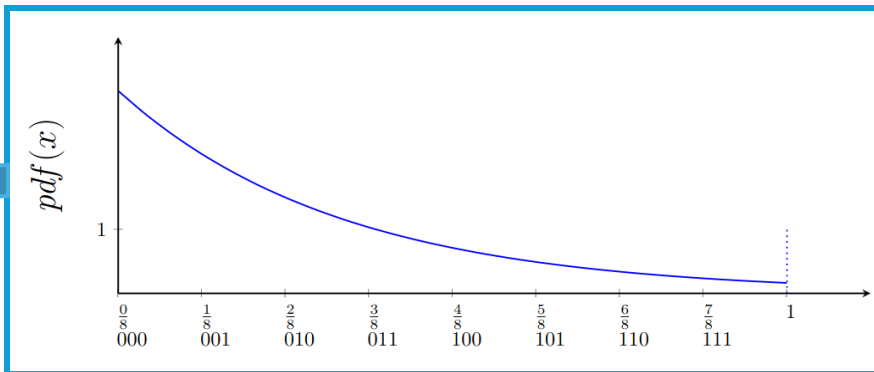
Essence of Bit Blasting



Which continuous distributions can be bit blasted?

Bit Blasting the Exponential $\lambda e^{-\lambda x}$

$$X \sim 0.b_1 b_2 b_3$$



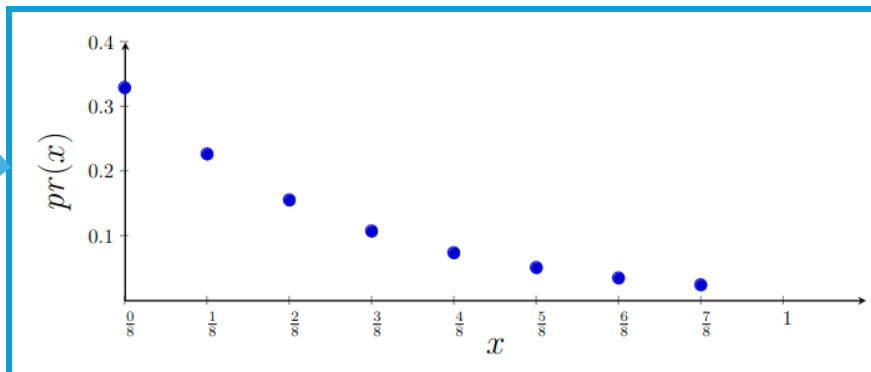
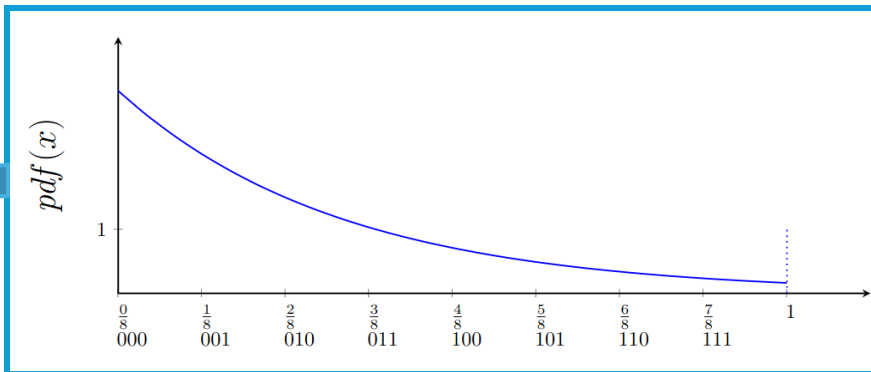
...

*represent bits using
a probabilistic program
of coin flips*

...

Bit Blasting the Exponential $\lambda e^{-\lambda x}$

$$X \sim 0.b_1b_2b_3$$



Bit Blast

`a = flip(0.1824)`

`b = flip(0.3208)`

`c = flip(0.4073)`

`[a, b, c]`

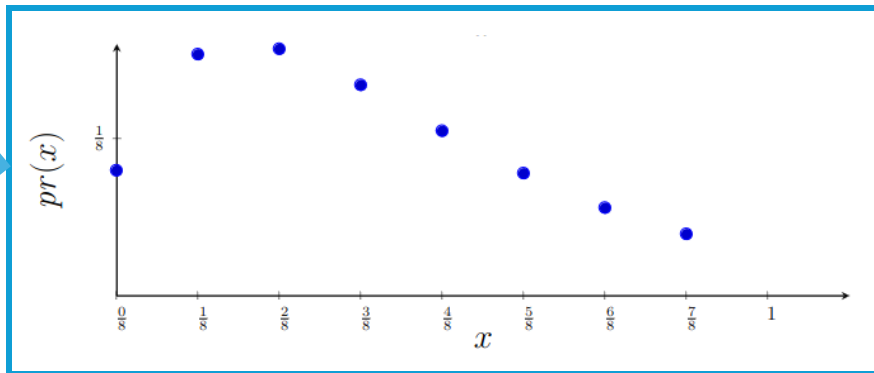
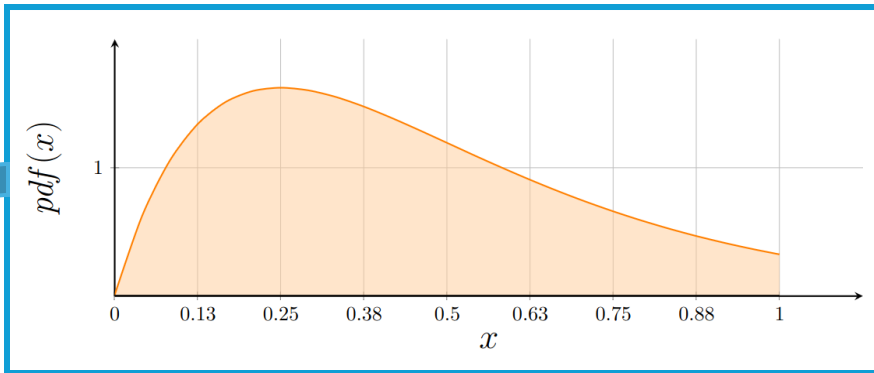
How many flips? k bits: k flips

Cannot go beyond the exponential with just independent coins!

Bit Blasting the Gamma

$$\frac{1}{\Gamma(k)\theta^k} x^{k-1} e^{-x/\theta}$$

$$X \sim 0.b_1b_2b_3$$



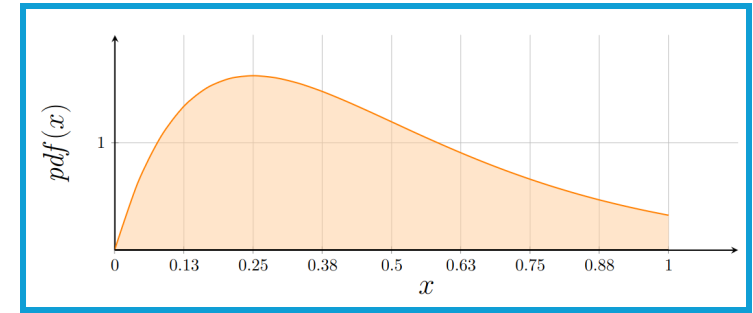
...

*represent bits using
a probabilistic program
of coin flips*

...

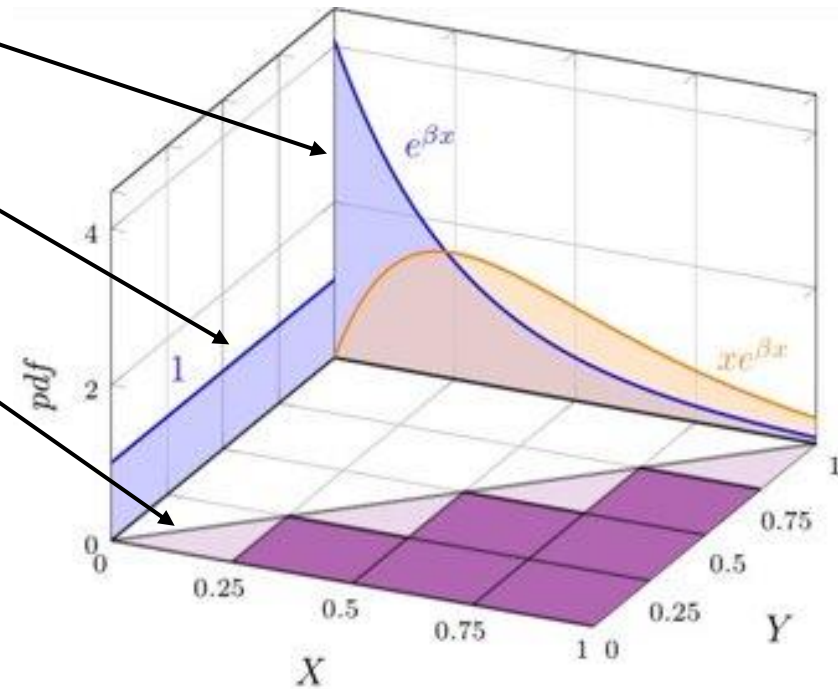


A Purely Continuous Gamma $x e^{-3x}$



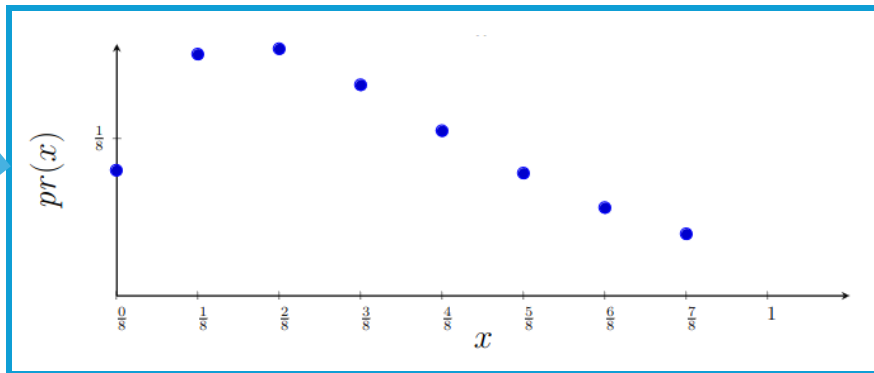
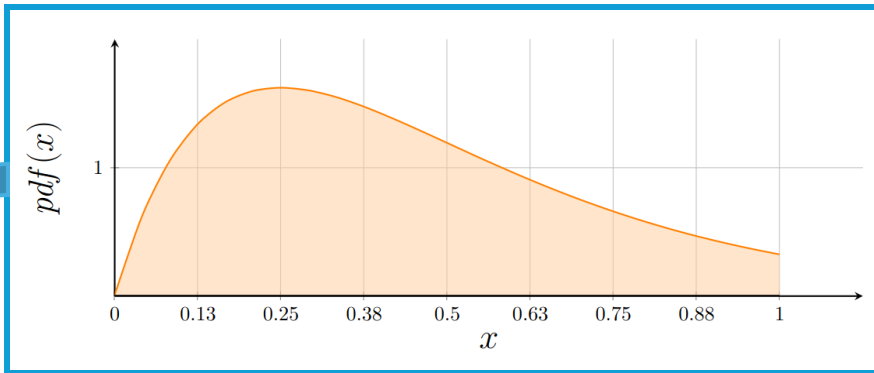
Continuous Program

```
X = exponential(-3)
Y = uniform(0, 1)
observe(Y < X)
return X
```



Bit Blasting the Gamma

$$X \sim 0.b_1b_2b_3$$



Bit Blast

```
X = bitblast(exponential(-3))
```

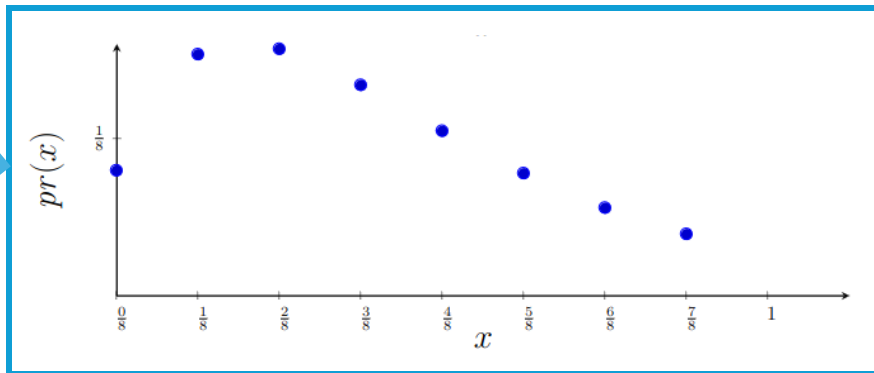
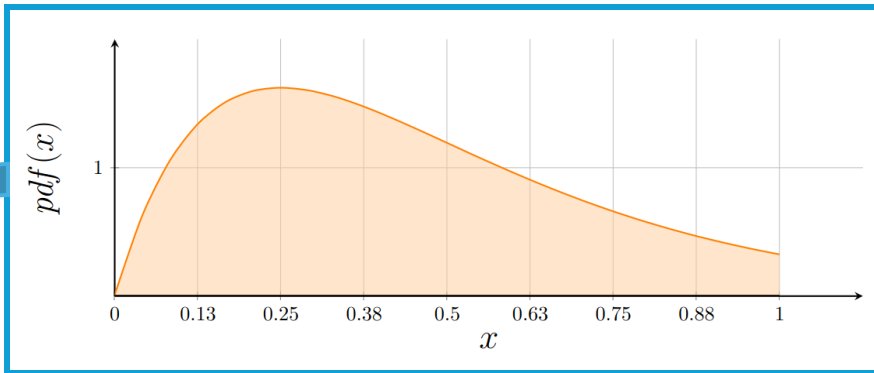
```
Y = bitblast(uniform(0, 1))
```

```
observe(Y < X)
```

```
return X
```

Bit Blasting the Gamma

$$X \sim 0.b_1b_2b_3$$



Bit Blast

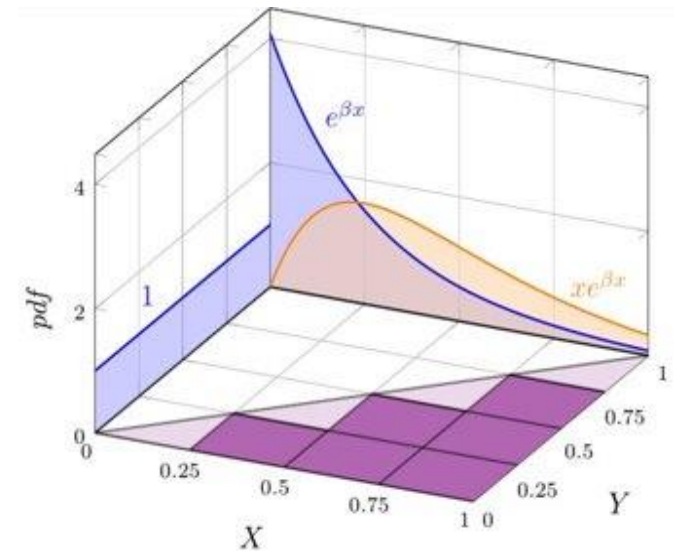
```
X = bitblast(exponential(-3))
```

```
Y = bitblast(uniform(0, 1))
```

```
observe(Y < X)
```

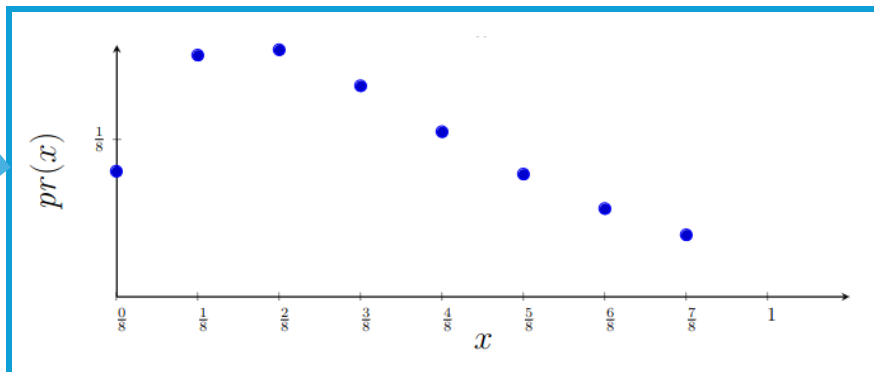
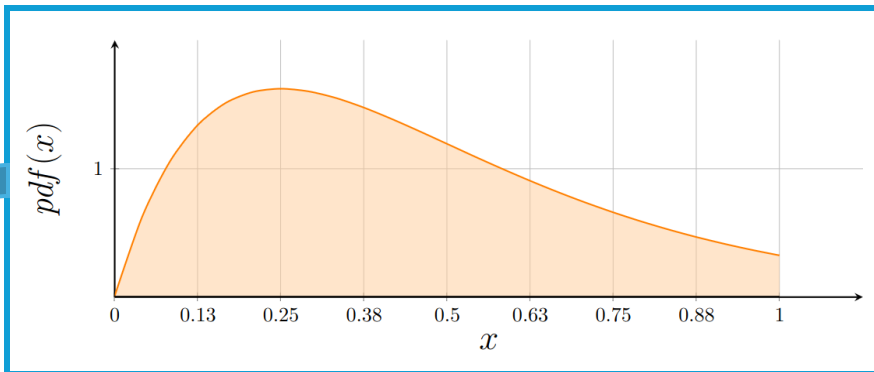
```
return X
```

Wrong



Bit Blasting the Gamma

$$X \sim 0. b_1 b_2 b_3$$



Bit Blast

```
X = [flip(.182), flip(.320), flip(.407)]  
Y = [flip(.5), flip(.5), flip(.5)]  
Z = [flip(.182), flip(.320), flip(.407)]
```

```
observe(Y < X)
```

```
return (if flip(0.208) then Z else X)
```

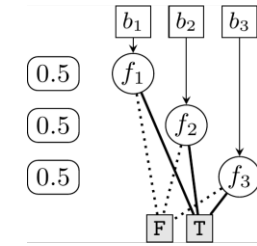
How many coin flips?

- 3 bits: 10 flips
- 32 bits: 97 flips
- k bits: $3k+1$ flips

Paper shows more:

- Efficient bit blasting for other common continuous distributions
- **HyBit** system for hybrid probabilistic programming <https://github.com/Tractable/Dice.jl/tree/hybit>
- Supports scalable probabilistic inference in Dice (core language guarantees BDDs of size $O(\text{poly}(k))$)
- Comprehensive evaluation on suite of hybrid programs:
 - HyBit supports all benchmarks
 - Gets the best accuracy on 11 out of 19 of them
- Check out our paper: <https://dl.acm.org/doi/10.1145/3656412>

Distribution	Density	Distribution	Density
Uniform	1	Gamma	$\frac{\beta^\alpha x^{\alpha-1} e^{-\beta x}}{\Gamma(\alpha)}$
Linear	x	Laplace	$\frac{1}{2b} e^{-\frac{ x-\mu }{b}}$
Polynomial	x^n	Chi-squared	$\frac{1}{2^{\frac{k}{2}} \Gamma(\frac{k}{2})} x^{\frac{k}{2}-1} e^{-\frac{x}{2}}$
Exponential	$\lambda e^{-\lambda x}$	Student-T	$c(1 + \frac{x^2}{v})^{-\frac{v+1}{2}}$



Benchmarks	HyBit		AQUA		WebPPL		Stan	CuBPTT
	Bit	Pieces	rejection	MCMC	SMC			
Pi [q10kdiver [n.d.]]	1.03E-04	14	✗	8.30E-05	9.66E-05	1.38E-03	4.84E-05	✗
weekend [Gehr et al. 2016]	2.08E-08	24	40%	✗	1.57E-02	1.57E-02	1.66E-02	✗
spaces [Luyten2019]	6.54E-04	19	42	✗	9.06E-04	3.24E-03	3.88E-02	✗
GPA [Wu et al. 2018]	2.22E-16	25	40%	3.62E-01	1.70E-02	9.39E-03	1.51E-02	✗
Tag of war [Huang et al. 2021]	4.58E-07	22	16	✗	6.93E-04	6.94E-04	2.33E-03	✗
altermez [Nishihara et al. 2013]	3.45E-06	17	25	3.41E-07	✗	6.41E-01	4.30E-01	✗
conjugate gaussians [Jordan 2010]	4.92E-06	23	16	0.99	2.19E-04	3.53E-04	3.18E-03	1.06E-04
normal_mix (p) [Huang et al. 2021]	5.49E-05	9	64	4.13E-07	✗	3.90E-04	5.30E-03	4.29E-01
normal_mix (mu) [Huang et al. 2021]	5.20E-03	9						1.87E+01
normal_mix (mu) [Huang et al. 2021]	3.92E-03	9						9.21E+00
zerone (w1) [Bissiri et al. 2016]	9.40E-05	16						1.77E+01
zerone (w2) [Bissiri et al. 2016]	4.51E-04	19						1.73E+01
coinflip [Gehr et al. 2016]	2.82E-07	22						1.18E+05
Addition [Gehr et al. 2016]	3.81E-06	23						8.45E+05
ClickGraph [Gehr et al. 2016]	1.75E-03	10						1.80E+05
truemill [Gehr et al. 2016]	3.65E-03	10						3.12E+02
clinicaltrial [Gehr et al. 2016]	5.27E-16	8						9.27E+04
clinicaltrial2 [Gehr et al. 2016]	6.81E-07	12						2.86E+01
additionmax [Gehr et al. 2016]	2.93E-07	23						1.19E+04

Bit Blasting Probabilistic Programs

POPOVA, GILLET, University of California, Los Angeles, USA
 GIL, YIN, USC, USC, University of California, Los Angeles, USA
 TODD, MILLSTEIN, University of California, Los Angeles, USA

Probabilistic programming languages (PPLs) are an expressive means for writing and reasoning about probabilistic models. They are used to model uncertainty in a wide range of applications, from medical diagnosis to financial risk analysis. However, PPLs are not yet widely used, in part due to their limited expressiveness and the difficulty of reasoning about their semantics. This paper introduces a new PPL, HyBit, that is designed to be both expressive and easy to reason about. HyBit is a hybrid PPL that combines the strengths of existing PPLs with a new, more expressive language for writing probabilistic models. HyBit is designed to be both expressive and easy to reason about. HyBit is a hybrid PPL that combines the strengths of existing PPLs with a new, more expressive language for writing probabilistic models. HyBit is designed to be both expressive and easy to reason about.