DESIGN AUTOMATION CONFERENCE

FROM CHIPS TO SYSTEMS — LEARN TODAY, CREATE TOMORROW

DEC 5 - 9, 2021 ◆ San Francisco, California

Statheros: A Compiler for Efficient Low-Precision Probabilistic Programming

Jacob Laurel, Rem Yang, Atharva Sehgal, Shubham Ugare, Sasa Misailovic

Department of Computer Science
University of Illinois at Urbana-Champaign

Probabilistic Programs

- Extend normal programs with :

Probabilistic Programs

- Extend normal programs with :

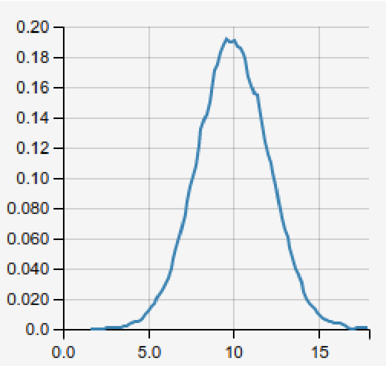
Random Sampling	`x |= Normal(0,1);`
Conditioning on Data	`Data<Fixed> y = {1.2,…}`
Posterior over Parameters	`Param<Fixed> x;`

Probabilistic Programs

Stan

UBER

PYRO



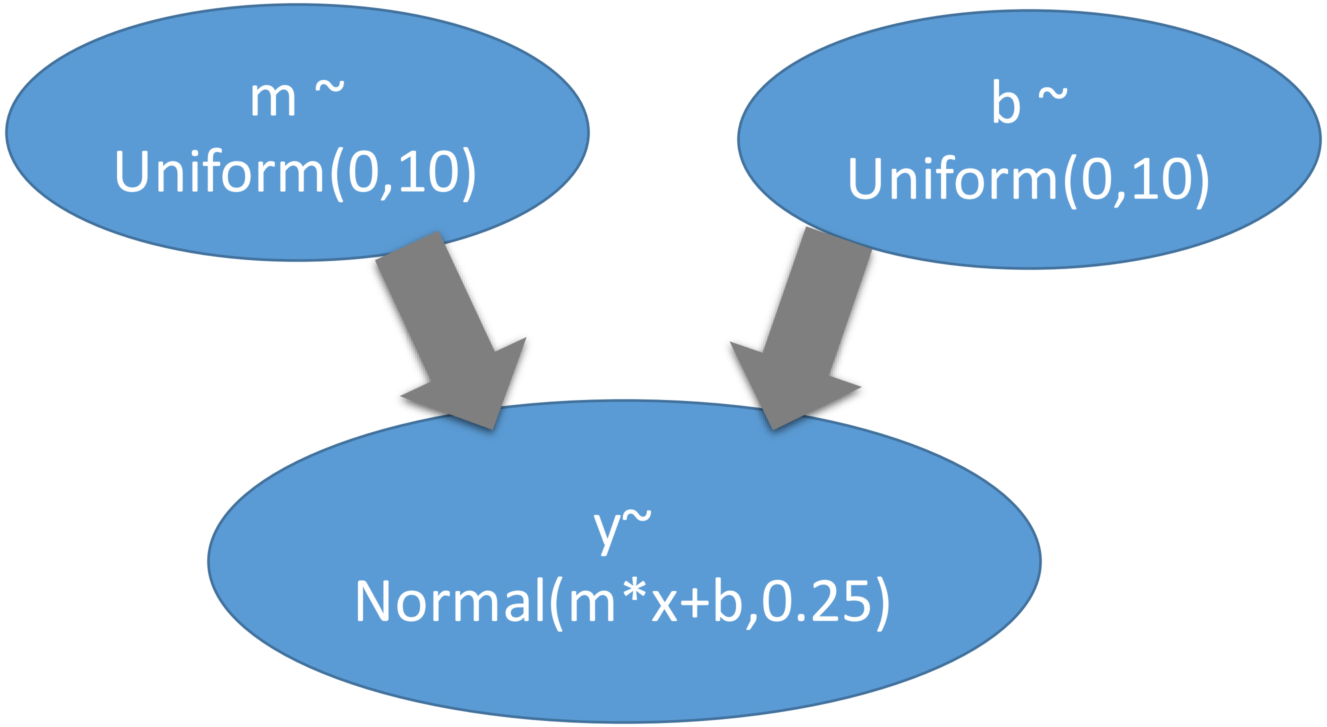
WebPPL probabilistic programming for the web

```
var gauss = function(){
    return sample(Gaussian({mu:  10, sigma: 2.1}))
}
```

XBOX LIVE

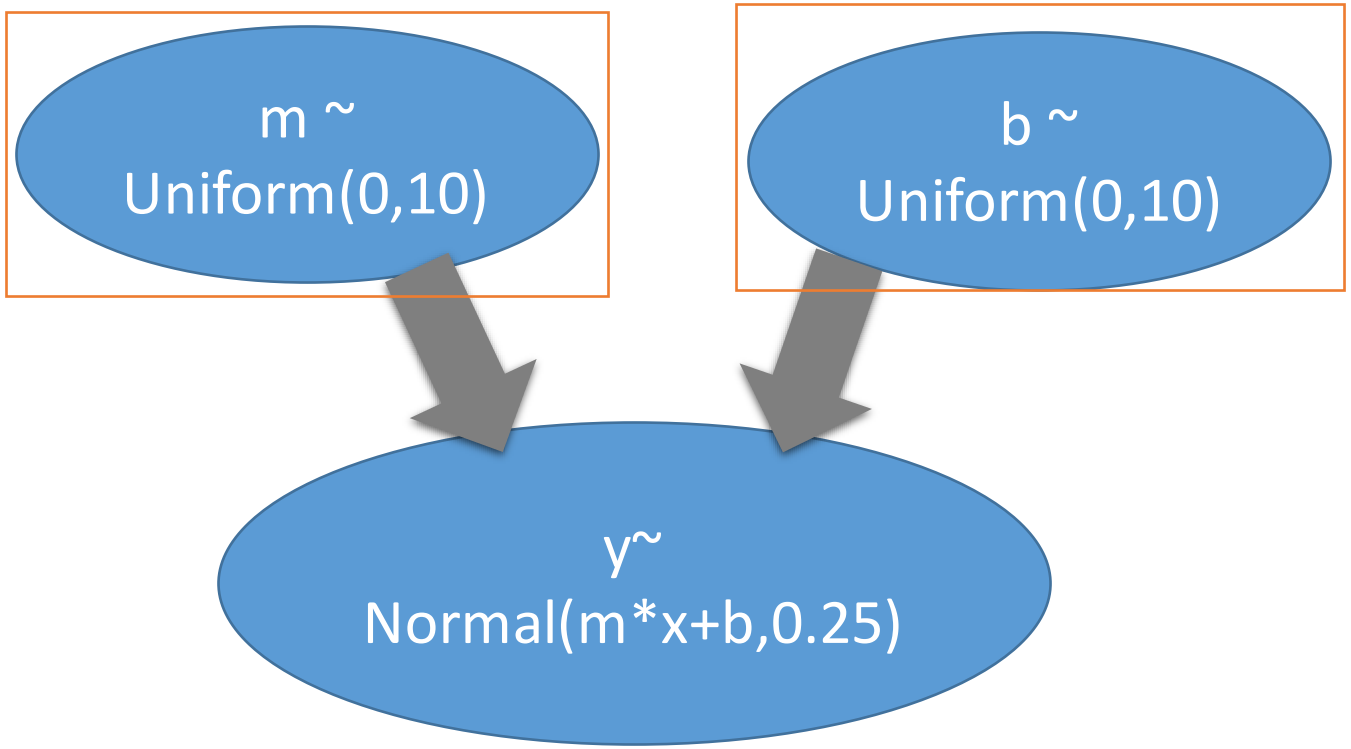
infer.net

Probabilistic Programs - Example




```
Param<Real> m;
Param<Real> b;
Data<Real> X = {1.2,…}
Data<Real> Y = {3.7,…}
m |= Uniform(0,10);
b |= Uniform(0,10);
Y |= Normal(m*X+b,0.25);
```

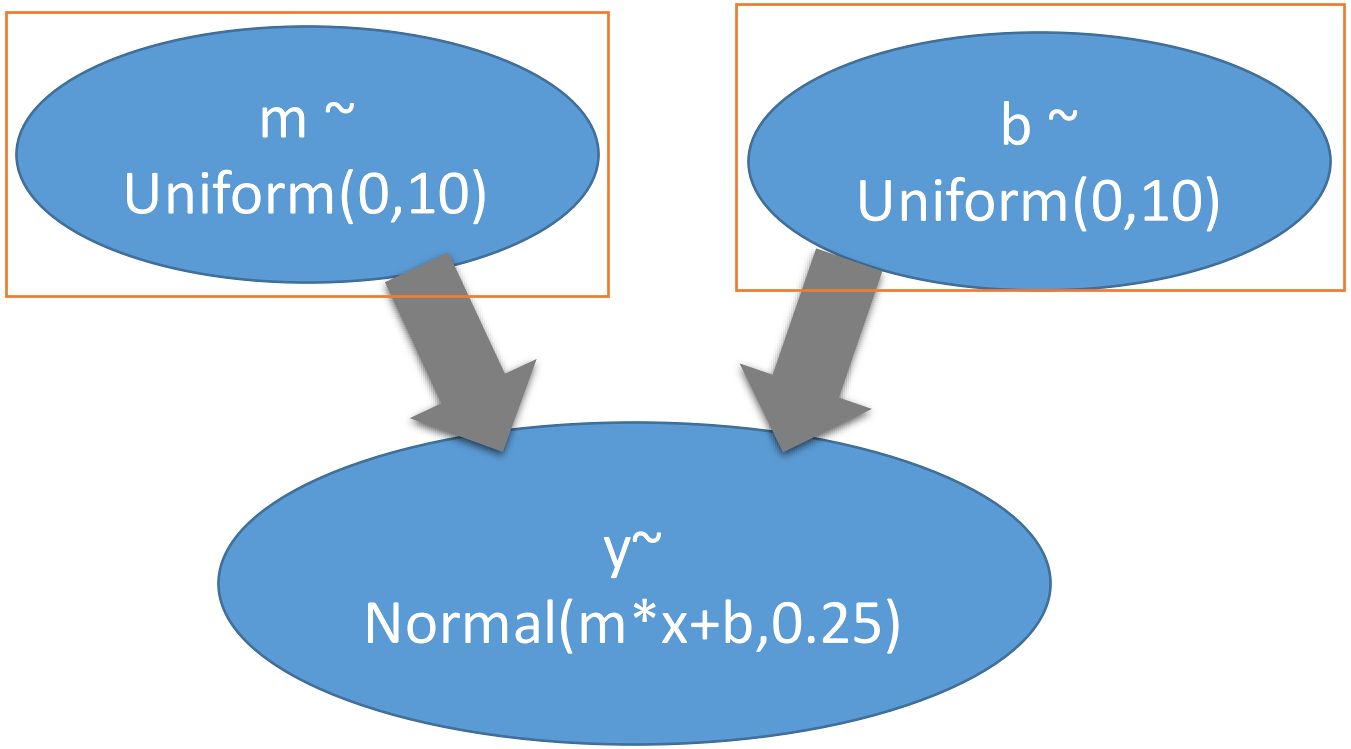
Probabilistic Programs - Example



Parameters to be inferred

```
Param<Real> m;
Param<Real> b;
Data<Real> X = {1.2,…}
Data<Real> Y = {3.7,…}
m |= Uniform(0,10);
b |= Uniform(0,10);
Y |= Normal(m*X+b,0.25);
```

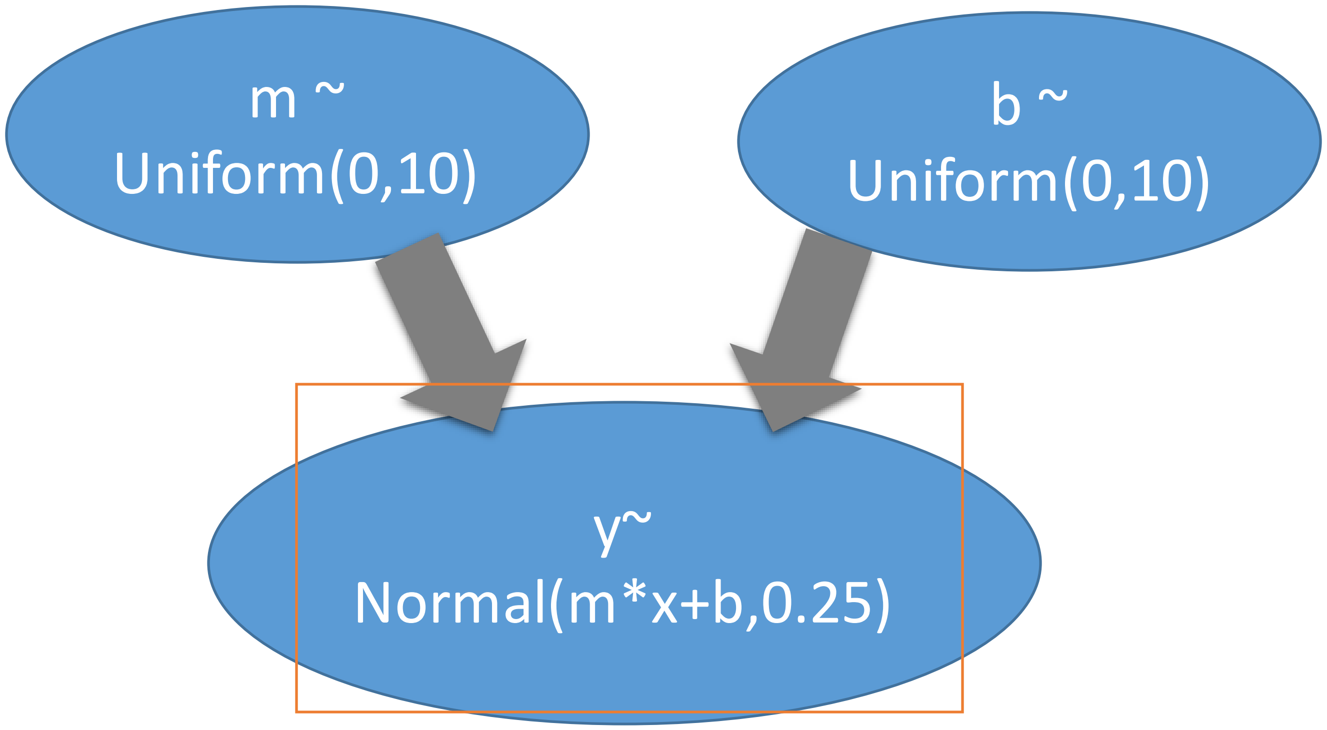
Probabilistic Programs - Example



Prior Distributions

```
Param<Real> m;
Param<Real> b;
Data<Real> X = {1.2,…}
Data<Real> Y = {3.7,…}
m  |= Uniform(0,10);
b  |= Uniform(0,10);
Y  |= Normal(m*X+b,0.25);
```

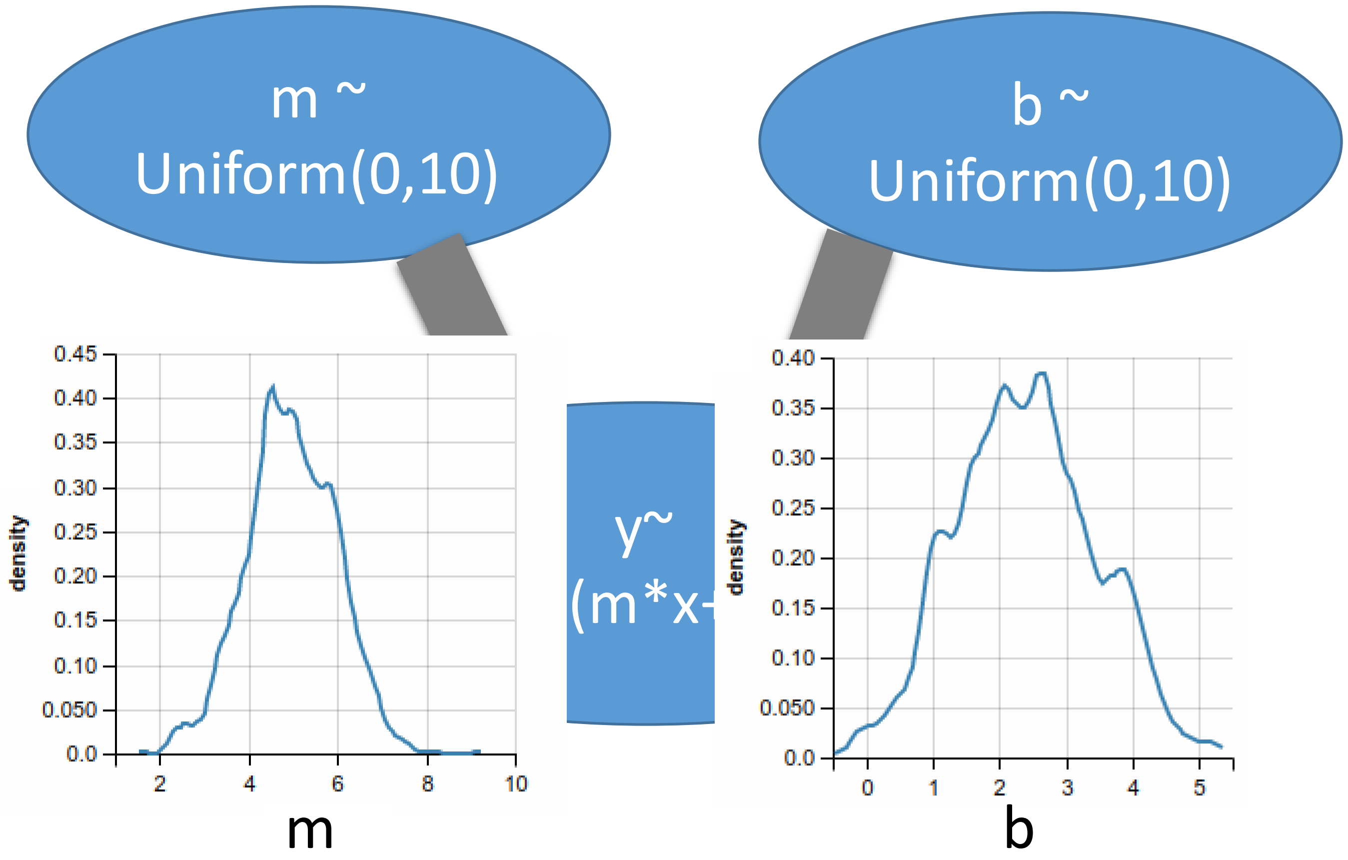
Probabilistic Programs - Example



Observed Data

```
Param<Real> m;
Param<Real> b;
Data<Real> X = {1.2,…}
Data<Real> Y = {3.7,…}
m |= Uniform(0,10);
b |= Uniform(0,10);
Y |= Normal(m*X+b,0.25);
```

Probabilistic Programs - Example



Posterior Distributions after Inference

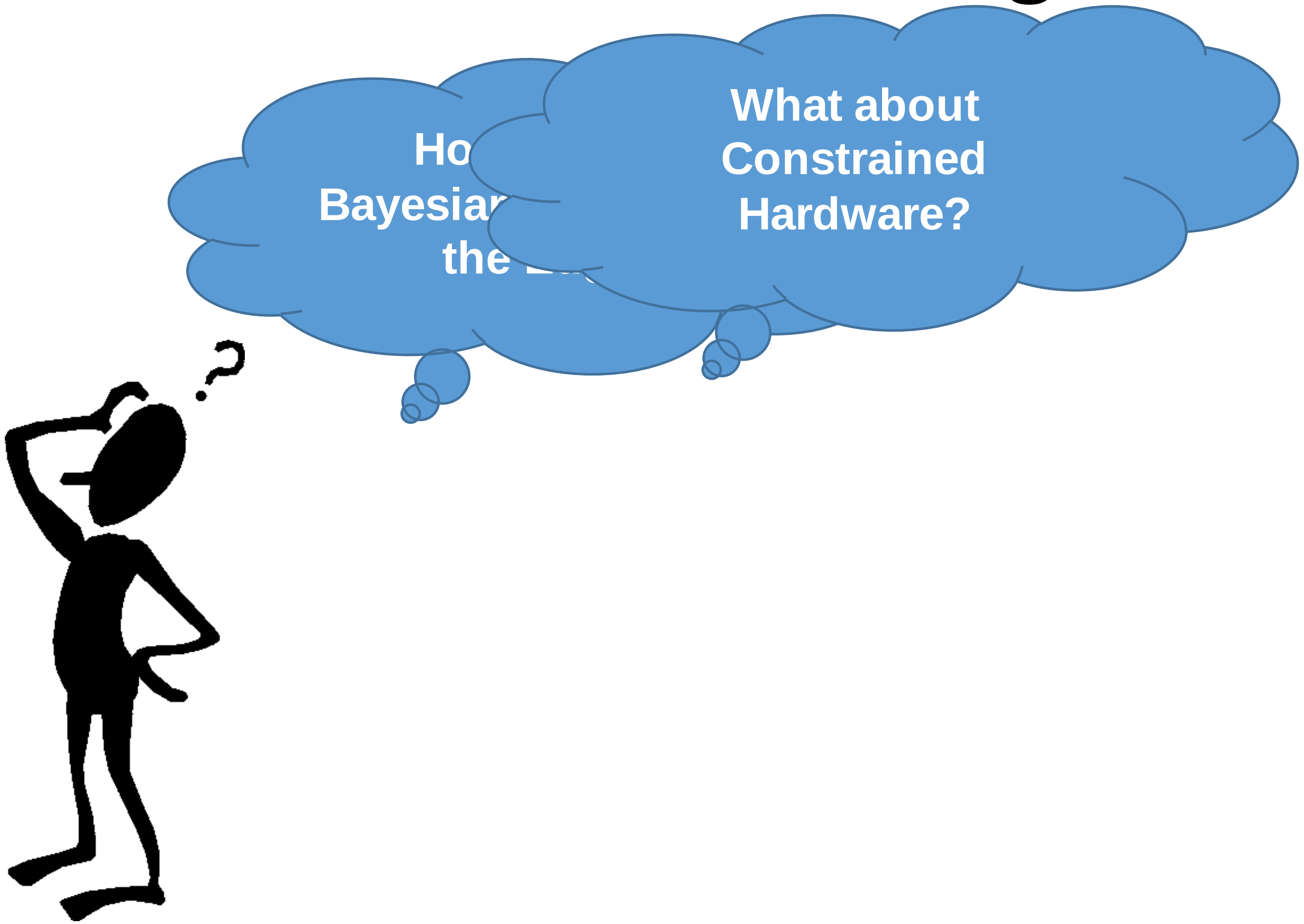
```
Param<Real> m;
Param<Real> b;
Data<Real> X = {1.2,…}
Data<Real> Y = {3.7,…}
m |= Uniform(0,10);
b |= Uniform(0,10);
Y |= Normal(m*X+b,0.25);
```

Edge Computing

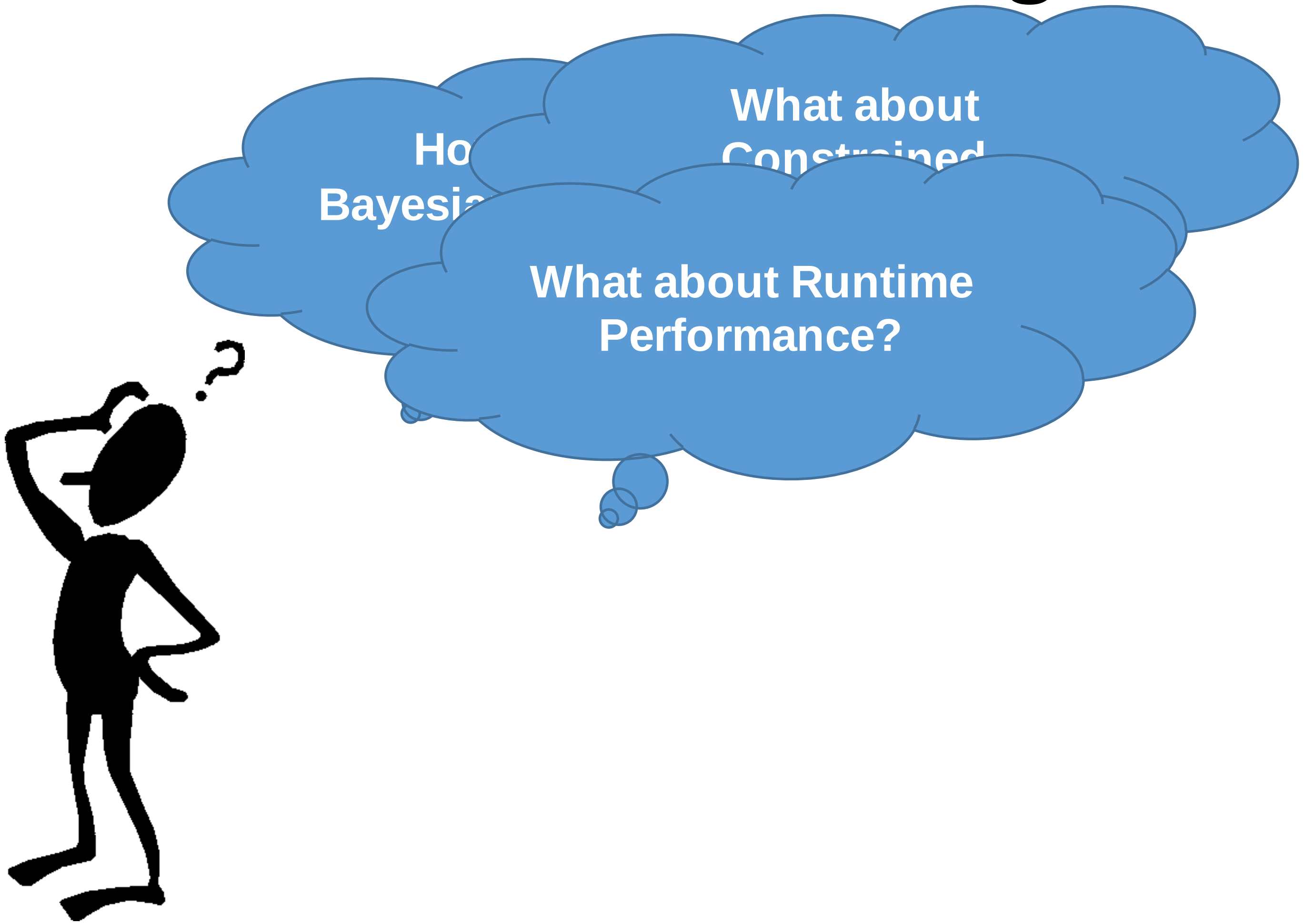
Bayesian Inference at the Edge

Bayesian Inference at the Edge

Bayesian Inference at the Edge



Bayesian Inference at the Edge



Bayesian Inference at the Edge

Idea 1)

Idea 1) Get a Ph.D. in ML + Embedded Systems

Idea 1) Get a Ph.D. in ML + Embedded Systems

Est. Time: 5-6 years

Idea 2)

Idea 2) Use fully automated Compiler framework

Statheros

- Embedded in C++
- All MCMC code uses fixed-point arithmetic
- Optimal configurations inferred by compiler
- Full integration with ARM toolchain for edge-device processors

Statheros Compiler Workflow

Statheros Compiler Workflow

```
Param<Real> m;
Param<Real> b;
Data<Real> X = {1.2,…}
Data<Real> Y = {3.7,…}
m |= Uniform(0,10);
b |= Uniform(0,10);
Y |= Normal(m*X+b,0.25)
```

Statheros Compiler Workflow

Fixed-Point Size Selection

```
Param<Real> m;
Param<Real> b;
Data<Real> X = {1.2,…}
Data<Real> Y = {3.7,…}
m |= Uniform(0,10);
b |= Uniform(0,10);
Y |= Normal(m*X+b,0.25)
```

Statheros Compiler Workflow

Fixed-Point Size Selection

Distribution Range Analysis

```
Param<Real> m;
Param<Real> b;
Data<Real> X = {1.2,…}
Data<Real> Y = {3.7,…}
m |= Uniform(0,10);
b |= Uniform(0,10);
Y |= Normal(m*X+b,0.25)
```

Statheros Compiler Workflow

Fixed-Point Size Selection

Distribution Range Analysis

MCMC Likelihood Range Analysis

```
Param<Real> m;
Param<Real> b;
Data<Real> X = {1.2,…}
Data<Real> Y = {3.7,…}
m |= Uniform(0,10);
b |= Uniform(0,10);
Y |= Normal(m*X+b,0.25)
```

Statheros Compiler Workflow

Fixed-Point Size Selection

Distribution Range Analysis

MCMC Likelihood Range Analysis

Instrumentation & Optimization

```
Param<Real> m;
Param<Real> b;
Data<Real> X = {1.2,…}
Data<Real> Y = {3.7,…}
m |= Uniform(0,10);
b |= Uniform(0,10);
Y |= Normal(m*X+b,0.25)
```

Statheros Compiler Workflow

Fixed-Point Size Selection

Distribution Range Analysis

MCMC Likelihood Range Analysis

Instrumentation & Optimization

Dynamic Checks

```
Param<Real> m;
Param<Real> b;
Data<Real> X = {1.2,…}
Data<Real> Y = {3.7,…}
m |= Uniform(0,10);
b |= Uniform(0,10);
Y |= Normal(m*X+b,0.25)
```

Statheros Compiler Workflow

Fixed-Point Size Selection

- Distribution Range Analysis
- MCMC Likelihood Range Analysis

Instrumentation & Optimization

- Dynamic Checks
- Domain-Specific Optimizations

```
Param<Real> m;
Param<Real> b;
Data<Real> X = {1.2,…}
Data<Real> Y = {3.7,…}
m |= Uniform(0,10);
b |= Uniform(0,10);
Y |= Normal(m*X+b,0.25)
```

Statheros Compiler Workflow

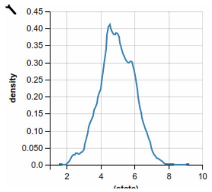
Fixed-Point Size Selection

- Distribution Range Analysis
- MCMC Likelihood Range Analysis

Instrumentation & Optimization

- Dynamic Checks
- Domain-Specific Optimizations

```
Param<Real> m;
Param<Real> b;
Data<Real> X = {1.2,…}
Data<Real> Y = {3.7,…}
m |= Uniform(0,10);
b |= Uniform(0,10);
Y |= Normal(m*X+b,0.25)
```



0x57 0x65 0x20 0x63 0x6f 0x6e 0x76
0x65 0x72 0x74 0x20 0x68 0x69 0x67
0x68 0x20 0x6c 0x65 0x76 0x65 0x6c
0x20 0x70 0x72 0x6f 0x62 0x61 0x62
0x69 0x6c 0x69 0x73 0x74 0x69 0x63
0x20 0x70 0x72 0x6f 0x67 0x72 0x61
0x6d 0x73 0x20 0x74 0x6f 0x20 0x6f
0x70 0x74 0x69 0x6d 0x69 0x7a 0x65
0x64 0x2c 0x20 0x63 0x6f 0x6d ...

Language Syntax

$Model$ ::= $Param^+$; $Data^+$; $DistStmt^+$;

$Param$ ::= `Param`<$Type$> (…)

$Data$ ::= `Data`<Type> (…)

$DistStmt$ ::= Var |= $DistExpr$ | Var |= $Expr$ | Var |= $Bexpr$? $Expr$: $Expr$
| `observe`($BExpr$) | `for (i=c1; i < c2; i++)` { Var`[i]` = $DistExpr$ }

$DistExpr$ ::= `bernoulli`($Expr$) | `uniform`($Expr$, $Expr$) | `normal`($Expr$, $Expr$)

$BExpr$ ::= $BExpr$ $Boolop$ $BExpr$ | $Expr$ $Relop$ $Expr$ | `true` | `false`

$Expr$ ::= $Expr$ $ArithOp$ $Expr$ | Var | c

$Type$ ::= `int` | `real` | `fixed<c,c>` | `vector`<$Type$>

$ArithOp \in \{+, -, *, /, **, \ldots\}$, $Boolop \in \{\, ||, \&\&, \ldots\}$, $Relop \in \{<, ==, <=, \ldots\}$

Language Syntax

Model ::= *Param*$^+$; *Data*$^+$; *DistStmt*$^+$;

Param ::= `Param`<*Type*> (…)

Data ::= `Data`<Type> (…)

DistStmt ::= *Var* |= *DistExpr* | *Var* |= *Expr* | *Var* |= *Bexpr* ? *Expr* : *Expr*
| `observe`(*BExpr*) | `for (i=c1; i < c2; i++)` { *Var*`[i]` = *DistExpr* }

DistExpr ::= `bernoulli`(*Expr*) | `uniform`(*Expr*, *Expr*) | `normal`(*Expr*, *Expr*)

BExpr ::= *BExpr* *Boolop* *BExpr* | *Expr* *Relop* *Expr* | `true` | `false`

Expr ::= *Expr* *ArithOp* *Expr* | *Var* | *c*

Type ::= `int` | `real` | `fixed<c,c>` | `vector`<*Type*>

ArithOp ∈ {+, -, *, /, **, …} , *Boolop* ∈ { ||, &&, …} , *Relop* ∈ {<, ==, <=, …}

Language Syntax

$Model$ ::= $Param^{+}$; $Data^{+}$; $DistStmt^{+}$;

$Param$ ::= $\texttt{Param}<Type>\texttt{(...)}$

$Data$ ::= $\texttt{Data}<\mathrm{Type}>\texttt{(...)}$

$DistStmt$ ::= Var |= $DistExpr$ | Var |= $Expr$ | Var |= $Bexpr$? $Expr$: $Expr$
| $\texttt{observe}(BExpr)$ | $\texttt{for (i=c1; i < c2; i++)}\ \{\ Var\texttt{[i]} = DistExpr\ \}$

$DistExpr$ ::= $\texttt{bernoulli}(Expr)$ | $\texttt{uniform}(Expr, Expr)$ | $\texttt{normal}(Expr, Expr)$

$BExpr$ ::= $BExpr\ Boolop\ BExpr$ | $Expr\ Relop\ Expr$ | $\texttt{true}$ | $\texttt{false}$

$Expr$ ::= $Expr\ ArithOp\ Expr$ | Var | c

$Type$ ::= $\texttt{int}$ | $\texttt{real}$ | $\texttt{fixed<c,c>}$ | $\texttt{vector}<Type>$

$ArithOp \in \{+, -, *, /, **, \ldots\}$, $Boolop \in \{ ||, \&\&, \ldots\}$, $Relop \in \{<, ==, <=, \ldots\}$

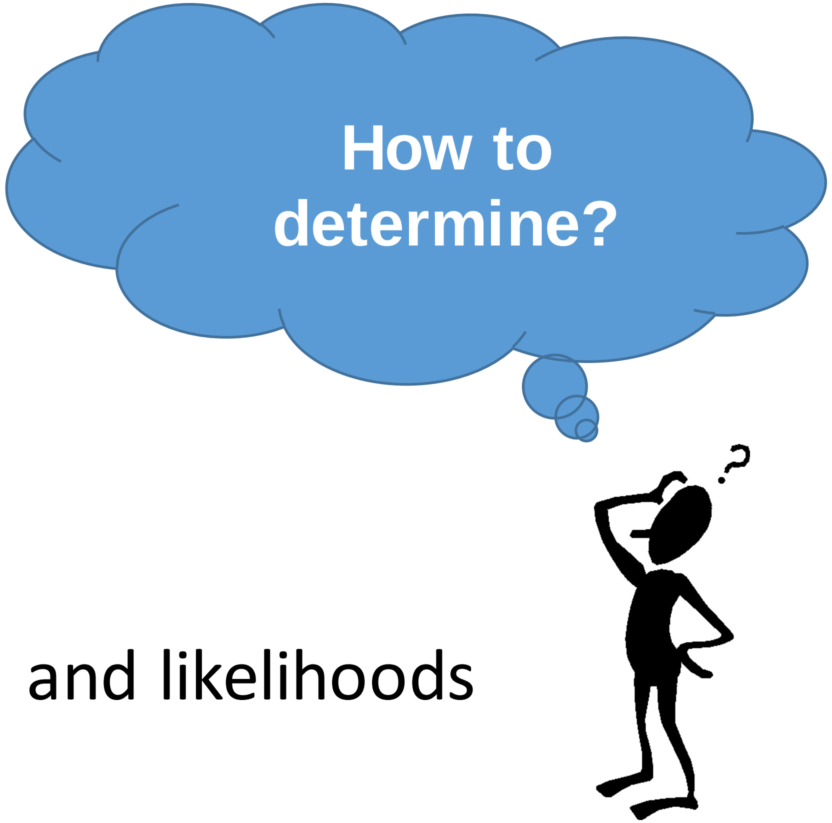
Step 1) Fixed-Point Size Selection

Fixed-Point Size Selection

- Fixed Point numbers given as <I,F>
- I is the amount of integer bits
- F is the amount of fractional bits
- Need enough integer bits for distributions and likelihoods

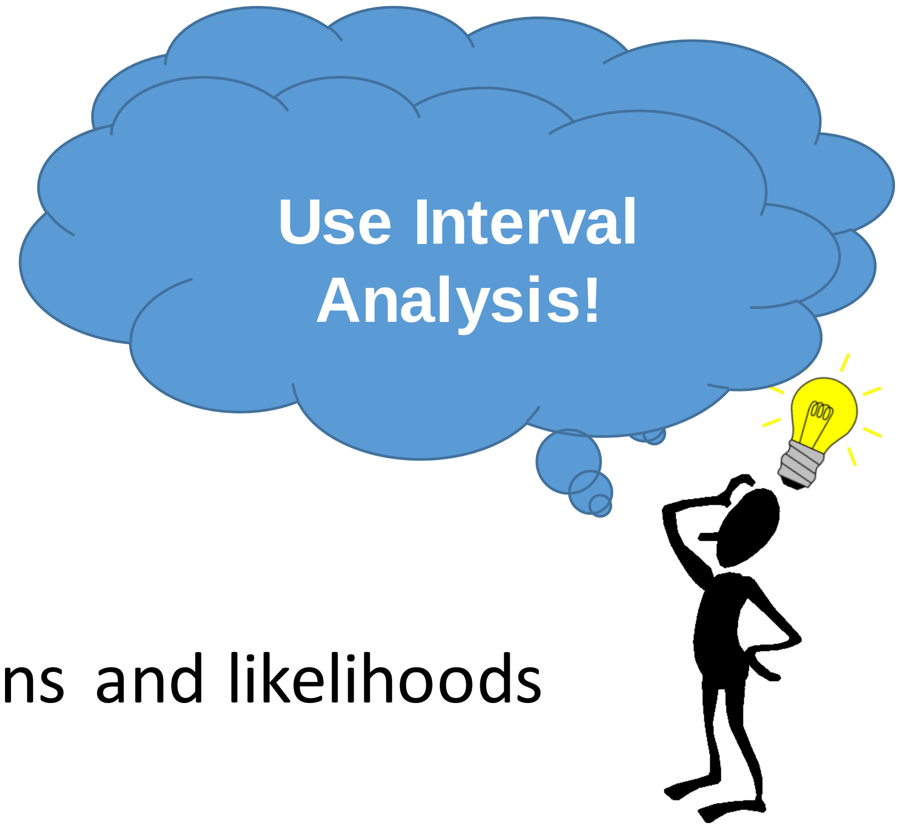
Fixed-Point Size Selection

- Fixed Point numbers given as <I,F>
- I is the amount of integer bits
- F is the amount of fractional bits
- Need <u>enough</u> integer bits for distributions and likelihoods



Fixed-Point Size Selection

- Fixed Point numbers given as <I,F>
- I is the amount of integer bits
- F is the amount of fractional bits
- Need <u>enough</u> integer bits for distributions and likelihoods



Fixed-Point Size Selection - Distributions

```
Param<Real> m;
Param<Real> b;
Data<Real> X = {1.2,2.4}
Data<Real> Y = {14.3,20.1}

m |= Uniform(0,10);
b |= Uniform(0,10);
Y |= Normal(m*X+b,0.25)
```

Fixed-Point Size Selection - Distributions

```
Param<Real> m;
Param<Real> b;
Data<Real> X = {1.2,2.4}
Data<Real> Y = {14.3,20.1}

m |= Uniform(0,10);
b |= Uniform(0,10);
Y |= Normal(m*X+b,0.25)
```

$X \in [1.2 , 2.4]$

Fixed-Point Size Selection - Distributions

```
Param<Real> m;
Param<Real> b;
Data<Real> X = {1.2,2.4}
Data<Real> Y = {14.3,20.1}
```

$Y \in [14.3 , 20.1]$

```
m |= Uniform(0,10);
b |= Uniform(0,10);
Y |= Normal(m*X+b,0.25)
```

Fixed-Point Size Selection - Distributions

```
Param<Real> m;
Param<Real> b;
Data<Real> X = {1.2,2.4}
Data<Real> Y = {14.3,20.1}

m |= Uniform(0,10);
b |= Uniform(0,10);
Y |= Normal(m*X+b,0.25)
```

$m \in [0 , 10]$

Fixed-Point Size Selection - Distributions

```
Param<Real> m;
Param<Real> b;
Data<Real> X = {1.2,2.4}
Data<Real> Y = {14.3,20.1}

m |= Uniform(0,10);
b |= Uniform(0,10);
Y |= Normal(m*X+b,0.25)
```

$b \in [0, 10]$

Fixed-Point Size Selection - Distributions

```
Param<Real> m;
Param<Real> b;
Data<Real> X = {1.2,2.4}
Data<Real> Y = {14.3,20.1}

m |= Uniform(0,10);
b |= Uniform(0,10);
Y |= Normal(m*X+b,0.25)
```

$Y \in [-\infty , \infty]$

Fixed-Point Size Selection - Distributions

```
Param<Real> m;
Param<Real> b;
Data<Real> X = {1.2,2.4}
Data<Real> Y = {14.3,20.1}
```

$Y \in [14.3 , 20.1]$

```
m |= Uniform(0,10);
b |= Uniform(0,10);
Y |= Normal(m*X+b,0.25)
```

$Y \in [-\infty , \infty]$

Fixed-Point Size Selection - Distributions

```
Param<Real> m;
Param<Real> b;
Data<Real> X = {1.2,2.4}
Data<Real> Y = {14.3,20.1}

m |= Uniform(0,10);
b |= Uniform(0,10);
Y |= Normal(m*X+b,0.25)
```

$Y \in [14.3 , 20.1]$

$\boxed{Y \in [-\infty , \infty]}$

Need to take bigger of the two

Fixed-Point Size Selection - Distributions

```
Param<Real> m;
Param<Real> b;
Data<Real> X = {1.2,2.4}
Data<Real> Y = {14.3,20.1}

m |= Uniform(0,10);
b |= Uniform(0,10);
Y |= Normal(m*X+b,0.25)
```

$Y \in [-\infty , \infty]$

Need to take bigger of the two

Can we do better?

Fixed-Point Size Selection - Distributions

```
Param<Real> m;
Param<Real> b;
Data<Real> X = {1.2,2.4}
Data<Real> Y = {14.3,20.1}

m |= Uniform(0,10);
b |= Uniform(0,10);
Y |= Normal(m*X+b,0.25)
```

$Y \in [-\infty , \infty]$

Need to take bigger of the two

Can we do better?

Yes!

Fixed-Point Size Selection - Distributions

```
Param<Real> m;
Param<Real> b;
Data<Real> X = {1.2,2.4}
Data<Real> Y = {14.3,20.1}

m |= Uniform(0,10);
b |= Uniform(0,10);
Y |= Normal(m*X+b,0.25)
```

$Y \in [-\infty , \infty]$

Samplers truncate to +/- 6σ

Fixed-Point Size Selection - Distributions

```
Param<Real> m;
Param<Real> b;
Data<Real> X = {1.2,2.4}
Data<Real> Y = {14.3,20.1}

m |= Uniform(0,10);
b |= Uniform(0,10);
Y |= Normal(m*X+b,0.25)
```

Samplers truncate to +/- 6σ

$Y \in [m*X+b-6*0.25 \; , \; m*X+b+6*0.25 \;]$

Fixed-Point Size Selection - Distributions

```
Param<Real> m;
Param<Real> b;
Data<Real> X = {1.2,2.4}
Data<Real> Y = {14.3,20.1}

m |= Uniform(0,10);
b |= Uniform(0,10);
Y |= Normal(m*X+b,0.25)
```

$Y \in [m*X+b-6*0.25 \ , \ m*X+b+6*0.25 \ ]$

Simplify using Interval Arithmetic

Fixed-Point Size Selection - Distributions

```
Param<Real> m;
Param<Real> b;
Data<Real> X = {1.2,2.4}
Data<Real> Y = {14.3,20.1}

m |= Uniform(0,10);
b |= Uniform(0,10);
Y |= Normal(m*X+b,0.25)
```

$Y \in [-1.5 , 35.5]$

Simplify using Interval Arithmetic

Fixed-Point Size Selection - Distributions

```
Param<Real> m;
Param<Real> b;
Data<Real> X = {1.2,2.4}
Data<Real> Y = {14.3,20.1}

m |= Uniform(0,10);
b |= Uniform(0,10);
Y |= Normal(m*X+b,0.25)
```

Fixed-Point Size Selection - Distributions

```
Param<Real> m;
Param<Real> b;
Data<Real> X = {1.2,2.4}
Data<Real> Y = {14.3,20.1}

m |= Uniform(0,10);
b |= Uniform(0,10);
Y |= Normal(m*X+b,0.25)
```

$X \in [1.2 , 2.4]$

$m \in [0 , 10]$

$b \in [0 , 10]$

$Y \in [-1.5 , 35.5]$

Final Intervals

Fixed-Point Size Selection - Likelihoods

- MCMC requires computing log-likelihoods for acceptance ratio
- Each distribution has different likelihood
- Need Summation over **all** data samples

Markov Chain Monte Carlo

$$\log(acc) = \sum_{i=1}^{m}(\log(\Pr(d_i \mid \mathbf{x_p})) + \log(\Pr(\mathbf{x_p})) + \log(q(\mathbf{x_c} \mid \mathbf{x_p})))$$
$$- \sum_{i=1}^{m}(\log(\Pr(d_i \mid \mathbf{x_c})) + \log(\Pr(\mathbf{x_c})) + \log(q(\mathbf{x_p} \mid \mathbf{x_c})))$$

Markov Chain Monte Carlo

Acceptance Ratio

$$\boxed{\log(acc)} = \sum_{i=1}^{m} \left(\log\left(\Pr(d_i \mid \mathbf{x_p})\right) + \log\left(\Pr(\mathbf{x_p})\right) + \log\left(q(\mathbf{x_c} \mid \mathbf{x_p})\right) \right)$$
$$- \sum_{i=1}^{m} \left(\log\left(\Pr(d_i \mid \mathbf{x_c})\right) + \log\left(\Pr(\mathbf{x_c})\right) + \log\left(q(\mathbf{x_p} \mid \mathbf{x_c})\right) \right)$$

Markov Chain Monte Carlo

Log-likelihoods

$$
\begin{aligned}
\log(acc) = & \sum_{i=1}^{m} \left(\boxed{\log(\Pr(d_i \mid \mathbf{x_p}))} + \boxed{\log(\Pr(\mathbf{x_p}))} + \boxed{\log(q(\mathbf{x_c} \mid \mathbf{x_p}))} \right) \\
& - \sum_{i=1}^{m} \left(\boxed{\log(\Pr(d_i \mid \mathbf{x_c}))} + \boxed{\log(\Pr(\mathbf{x_c}))} + \boxed{\log(q(\mathbf{x_p} \mid \mathbf{x_c}))} \right)
\end{aligned}
$$

Markov Chain Monte Carlo

Summation over **all** observed data

$$\log(acc) = \boxed{\sum_{i=1}^{m}}\left(\log\left(\Pr(d_i \mid \mathbf{x_p})\right) + \log\left(\Pr(\mathbf{x_p})\right) + \log\left(q(\mathbf{x_c} \mid \mathbf{x_p})\right)\right)$$

$$\boxed{-\sum_{i=1}^{m}}\left(\log\left(\Pr(d_i \mid \mathbf{x_c})\right) + \log\left(\Pr(\mathbf{x_c})\right) + \log\left(q(\mathbf{x_p} \mid \mathbf{x_c})\right)\right)$$

Fixed-Point Size Selection - Likelihoods

- How to bound terms like $(\log\left(\Pr(d_i \mid \mathbf{x}_c)\right)$ and $\log\left(\Pr(\mathbf{x}_c)\right)$?

Fixed-Point Size Selection - Likelihoods

- How to bound terms like $(\log(\Pr(d_i \mid \mathbf{x_c}))$ and $\log(\Pr(\mathbf{x_c}))$?
- Propagate interval bounds through each distribution's likelihood

Fixed-Point Size Selection - Likelihoods

- How to bound terms like $(\log\left(\Pr(d_i \,|\, \mathbf{x_c})\right)$ and $\log\left(\Pr(\mathbf{x_c})\right)$?
- Propagate interval bounds through each distribution's <u>likelihood</u>
- Leverage previously computed intervals!

Fixed-Point Size Selection - Likelihoods

Fixed-Point Size Selection - Likelihoods

```
Param<Real> m;
Param<Real> b;
Data<Real> X = {1.2,2.4}
Data<Real> Y = {14.3,20.1}

m |= Uniform(0,10);
b |= Uniform(0,10);
Y |= Normal(m*X+b,0.25)
```

Fixed-Point Size Selection - Likelihoods

```
Param<Real> m;
Param<Real> b;
Data<Real> X = {1.2,2.4}
Data<Real> Y = {14.3,20.1}

m |= Uniform(0,10);
b |= Uniform(0,10);
Y |= Normal(m*X+b,0.25)
```

Computing: $\log\left(\Pr(\mathbf{x_c})\right)$

Log-Likelihood[m] $\in \left[\log\left(\frac{1}{10}\right), \log\left(\frac{1}{10}\right)\right]$

Fixed-Point Size Selection - Likelihoods

Computing: $\log\left(\Pr(\mathbf{x_c})\right)$

```
Param<Real> m;
Param<Real> b;
Data<Real> X = {1.2,2.4}
Data<Real> Y = {14.3,20.1}

m |= Uniform(0,10);
b |= Uniform(0,10);
Y |= Normal(m*X+b,0.25)
```

Log-Likelihood[b] $\in \left[\log\left(\frac{1}{10}\right), \log\left(\frac{1}{10}\right)\right]$

Fixed-Point Size Selection - Likelihoods

```
Param<Real> m;
Param<Real> b;
Data<Real> X = {1.2,2.4}
Data<Real> Y = {14.3,20.1}

m |= Uniform(0,10);
b |= Uniform(0,10);
Y |= Normal(m*X+b,0.25)
```

Computing: $(\log(\Pr(d_i \mid \mathbf{x_c}))$

Log-Likelihood[Y] $\in$???

Fixed-Point Size Selection - Likelihoods

```
Param<Real> m;
Param<Real> b;
Data<Real> X = {1.2,2.4}
Data<Real> Y = {14.3,20.1}

m |= Uniform(0,10);
b |= Uniform(0,10);
Y |= Normal(m*X+b,0.25)
```

Leverage previously computed intervals!

Log-Likelihood[Y] ∊ ???

Fixed-Point Size Selection - Likelihoods

```
Param<Real> m;
Param<Real> b;
Data<Real> X = {1.2,2.4}
Data<Real> Y = {14.3,20.1}

m |= Uniform(0,10);
b |= Uniform(0,10);
Y |= Normal(m*X+b,0.25)
```

X ∊ [1.2 , 2.4]

m ∊ [0 , 10]

b ∊ [0 , 10]

Y ∊ [-1.5 , 35.5]

Log-Likelihood[Y] ∊

Fixed-Point Size Selection - Likelihoods

```
Param<Real> m;
Param<Real> b;
Data<Real> X = {1.2,2.4}
Data<Real> Y = {14.3,20.1}

m |= Uniform(0,10);
b |= Uniform(0,10);
Y |= Normal(m*X+b,0.25)
```

$X \in [1.2 , 2.4]$

$m \in [0 , 10]$

$b \in [0 , 10]$

$Y \in [-1.5 , 35.5]$

$m*X+b \in [0 , 34]$

Log-Likelihood[Y] $\in$

Fixed-Point Size Selection - Likelihoods

```
Param<Real> m;
Param<Real> b;
Data<Real> X = {1.2,2.4}
Data<Real> Y = {14.3,20.1}

m |= Uniform(0,10);
b |= Uniform(0,10);
Y |= Normal(m*X+b,0.25)
```

$X \in [1.2 , 2.4]$

$m \in [0 , 10]$

$b \in [0 , 10]$

$Y \in [-1.5 , 35.5]$

$m*X+b \in [0 , 34]$

Log-Likelihood[Y] $\in$

$$\log\left(\frac{1}{\sqrt{2\pi}[0.25,0.25]}\right) - \frac{1}{2}\left(\frac{[-1.5,35.5]-[0,34]}{[0.25,0.25]}\right)^2$$

Fixed-Point Size Selection - Likelihoods

```
Param<Real> m;
Param<Real> b;
Data<Real> X = {1.2,2.4}
Data<Real> Y = {14.3,20.1}

m |= Uniform(0,10);
b |= Uniform(0,10);
Y |= Normal(m*X+b,0.25)
```

$X \in [1.2 , 2.4]$

$m \in [0 , 10]$

$b \in [0 , 10]$

$Y \in [-1.5 , 35.5]$

$m*X+b \in [0 , 34]$

Log-Likelihood[Y] $\in$ [-10080.4 , 1.595]

$$\log\left(\frac{1}{\sqrt{2\pi}[0.25,0.25]}\right) - \frac{1}{2}\left(\frac{[-1.5,35.5]-[0,34]}{[0.25,0.25]}\right)^2$$

Fixed-Point Size Selection - Proposals

- Still need to bound proposal kernel terms:

$$\log\left(q(\mathbf{x_p} \mid \mathbf{x_c})\right) \quad \& \quad \log\left(q(\mathbf{x_c} \mid \mathbf{x_p})\right)$$

Fixed-Point Size Selection - Proposals

- Still need to bound proposal kernel terms:

$$\log\left(q(\mathbf{x_p} \mid \mathbf{x_c})\right)) \quad \& \quad \log\left(q(\mathbf{x_c} \mid \mathbf{x_p})\right)$$

- Proposal kernel has known form (Normal, uniform, etc.)

- $\mathbf{x_p}$ and $\mathbf{x_c}$ have known non-infinite interval bounds

- Proposal is **symmetric**

Fixed-Point Size Selection - Proposals

Computing: $\log\left(q(\mathbf{x_p} \mid \mathbf{x_c})\right))$

```
Param<Real> m;
Param<Real> b;
Data<Real> X = {1.2,2.4}
Data<Real> Y = {14.3,20.1}

m |= Uniform(0,10);
b |= Uniform(0,10);
Y |= Normal(m*X+b,0.25)
```

When proposal kernel is Normal($\mathbf{x_c}$,1)

Fixed-Point Size Selection - Proposals

Computing: $\log\left(q(\mathbf{x_p} \mid \mathbf{x_c})\right))$

```
Param<Real> m;
Param<Real> b;
Data<Real> X = {1.2,2.4}
Data<Real> Y = {14.3,20.1}

m |= Uniform(0,10);
b |= Uniform(0,10);
Y |= Normal(m*X+b,0.25)
```

When proposal kernel is Normal($\mathbf{x_c}$,1)

$m \in [0\ ,\ 10]$
$b \in [0\ ,\ 10]$

Fixed-Point Size Selection - Proposals

Computing: $\log\left(q(\mathbf{x_p} \mid \mathbf{x_c})\right))$

```
Param<Real> m;
Param<Real> b;
Data<Real> X = {1.2,2.4}
Data<Real> Y = {14.3,20.1}

m |= Uniform(0,10);
b |= Uniform(0,10);
Y |= Normal(m*X+b,0.25)
```

When proposal kernel is Normal($\mathbf{x_c}$,1)

$m \in [0\ ,\ 10]$

$b \in [0\ ,\ 10]$

Log-Likelihood[m]= Log-Likelihood[b] $\in$

$$\log\left(\frac{1}{\sqrt{2\pi}[1,1]}\right) - \frac{1}{2}\left(\frac{[0,10]-[0,10]}{[1,1]}\right)^2$$

Fixed-Point Size Selection - Proposals

Computing: $\log\left(q(\mathbf{x_p} \mid \mathbf{x_c})\right)$

```
Param<Real> m;
Param<Real> b;
Data<Real> X = {1.2,2.4}
Data<Real> Y = {14.3,20.1}

m |= Uniform(0,10);
b |= Uniform(0,10);
Y |= Normal(m*X+b,0.25)
```

When proposal kernel is Normal($\mathbf{x_c}$,1)

m ∊ [0 , 10]
b ∊ [0 , 10]

Log-Likelihood[m]= Log-Likelihood[b] ∊

$$\log\left(\frac{1}{\sqrt{2\pi}[1,1]}\right) - \frac{1}{2}\left(\frac{[0,10]-[0,10]}{[1,1]}\right)^2$$

Fixed-Point Size Selection - Proposals

Computing: $\log\left(q(\mathbf{x_p} \mid \mathbf{x_c})\right)$

```
Param<Real> m;
Param<Real> b;
Data<Real> X = {1.2,2.4}
Data<Real> Y = {14.3,20.1}

m |= Uniform(0,10);
b |= Uniform(0,10);
Y |= Normal(m*X+b,0.25)
```

When proposal kernel is Normal($\mathbf{x_c}$,1)

$m \in [0\ ,\ 10]$

$b \in [0\ ,\ 10]$

Log-Likelihood[m]= Log-Likelihood[b] ∊ [-50.9, -0.91]

$$\log\left(\frac{1}{\sqrt{2\pi}[1,1]}\right) - \frac{1}{2}\left(\frac{[0,10]-[0,10]}{[1,1]}\right)^2$$

Fixed-Point Size Selection - MCMC

- We can now bound all terms for

$$\log\left(acc\right) = \sum_{i=1}^{m}\left(\log\left(\Pr(d_i \mid \mathbf{x_p})\right) + \log\left(\Pr(\mathbf{x_p})\right) + \log\left(q(\mathbf{x_c} \mid \mathbf{x_p})\right)\right)$$
$$- \sum_{i=1}^{m}\left(\log\left(\Pr(d_i \mid \mathbf{x_c})\right) + \log\left(\Pr(\mathbf{x_c})\right) + \log\left(q(\mathbf{x_p} \mid \mathbf{x_c})\right)\right)$$

Fixed-Point Size Selection - MCMC

- We can now bound all terms for

$$\log\left(acc\right) = \sum_{i=1}^{m}\left(\boxed{\log\left(\Pr(d_i \mid \mathbf{x_p})\right)} + \log\left(\Pr(\mathbf{x_p})\right) + \log\left(q(\mathbf{x_c} \mid \mathbf{x_p})\right)\right)$$
$$- \sum_{i=1}^{m}\left(\boxed{\log\left(\Pr(d_i \mid \mathbf{x_c})\right)} + \log\left(\Pr(\mathbf{x_c})\right) + \log\left(q(\mathbf{x_p} \mid \mathbf{x_c})\right)\right)$$

$$[-10080.4, 1.595]$$

Fixed-Point Size Selection - MCMC

- We can now bound all terms for

$$\log\left(acc\right) = \textstyle\sum_{i=1}^{m} (\log\left(\Pr(d_i \mid \mathbf{x_p})\right) + \boxed{\log\left(\Pr(\mathbf{x_p})\right)} + \log\left(q(\mathbf{x_c} \mid \mathbf{x_p})\right))$$
$$- \textstyle\sum_{i=1}^{m} (\log\left(\Pr(d_i \mid \mathbf{x_c})\right) + \boxed{\log\left(\Pr(\mathbf{x_c})\right)} + \log\left(q(\mathbf{x_p} \mid \mathbf{x_c})\right))$$

$$\left[\log\left(\tfrac{1}{10}\right), \log\left(\tfrac{1}{10}\right)\right] + \left[\log\left(\tfrac{1}{10}\right), \log\left(\tfrac{1}{10}\right)\right]$$

Fixed-Point Size Selection - MCMC

- We can now bound all terms for

$$\log(acc) = \sum_{i=1}^{m} (\log(\Pr(d_i \mid \mathbf{x_p})) + \log(\Pr(\mathbf{x_p})) + \boxed{\log(q(\mathbf{x_c} \mid \mathbf{x_p}))})$$

$$- \sum_{i=1}^{m} (\log(\Pr(d_i \mid \mathbf{x_c})) + \log(\Pr(\mathbf{x_c})) + \boxed{\log(q(\mathbf{x_p} \mid \mathbf{x_c})))}$$

$$[-50.9, -0.91]$$

Fixed-Point Size Selection - MCMC

- We can now bound all terms for

$$\log(acc) = \sum_{i=1}^{m} \boxed{\left(\log\left(\Pr(d_i \mid \mathbf{x_p})\right) + \log\left(\Pr(\mathbf{x_p})\right) + \log\left(q(\mathbf{x_c} \mid \mathbf{x_p})\right)\right)}$$
$$- \sum_{i=1}^{m} \boxed{\left(\log\left(\Pr(d_i \mid \mathbf{x_c})\right) + \log\left(\Pr(\mathbf{x_c})\right) + \log\left(q(\mathbf{x_p} \mid \mathbf{x_c})\right)\right)}$$

$$[-10135.50, -3.92]$$

Fixed-Point Size Selection - MCMC

- We can now bound all terms for

$$\log(acc) = \boxed{\sum_{i=1}^{m}}\left(\log\left(\Pr(d_i \mid \mathbf{x_p})\right) + \log\left(\Pr(\mathbf{x_p})\right) + \log\left(q(\mathbf{x_c} \mid \mathbf{x_p})\right)\right)$$
$$- \boxed{\sum_{i=1}^{m}}\left(\log\left(\Pr(d_i \mid \mathbf{x_c})\right) + \log\left(\Pr(\mathbf{x_c})\right) + \log\left(q(\mathbf{x_p} \mid \mathbf{x_c})\right)\right)$$

$$[-10135.50, -3.92]$$

$*$ Number of Observations

Can we get away with less?

Can we get away with less?

Yes!

Can we get away with less?

Yes!   Overflows are ok.

Fixed-Point Size Selection – Overflows

- Fixed-Point uses **2's complement** integer arithmetic

Fixed-Point Size Selection – Overflows

- Fixed-Point uses **2's complement** integer arithmetic
- Wrap-around overflows in Likelihood Summation are **ok...**

Fixed-Point Size Selection – Overflows

- Fixed-Point uses **2's complement** integer arithmetic

- Wrap-around overflows in Likelihood Summation are **ok...**

  ...provided **final** result is within representable range

Fixed-Point Size Selection – Overflows

- Fixed-Point uses **2's complement** integer arithmetic

- Wrap-around overflows in Likelihood Summation are **ok...**

  ...provided **final** result is within representable range

→ Only need enough integer bits for largest **single** likelihood

Fixed-Point Size Selection

- Given interval bounds how do we choose final size?

Fixed-Point Size Selection

- Given interval bounds how do we choose final size?
- Need a different size for distributions and likelihoods!

Fixed-Point Size Selection

- Given interval bounds how do we choose final size?

- Need a different size for <u>distributions</u> and likelihoods!

$$I_M \geq \log_2 \left(\left\lceil \max_{v \in Vars}(|\underline{v}|, |\overline{v}|) \right\rceil \right)$$ Integer Bits

Fixed-Point Size Selection

- Given interval bounds how do we choose final size?

- Need a different size for <u>distributions</u> and likelihoods!

$$I_M \geq \log_2 \left(\left\lceil \max_{v \in Vars}(|\underline{v}|, |\overline{v}|) \right\rceil \right)$$

$$F_M = 32 - 1 - I_M$$ Fractional Bits

Fixed-Point Size Selection

- Given interval bounds how do we choose final size?

- Need a different size for distributions and likelihoods!

$$I_{LL} \geq \log_2\left(\left\lceil \max_{v \in Vars}(|\text{log-likelihood}(\underline{v})|, |\text{log-likelihood}(\overline{v})|) \right\rceil\right)$$

Fixed-Point Size Selection

- Given interval bounds how do we choose final size?
- Need a different size for distributions and <u>likelihoods</u>!

$$I_{LL} \geq \log_2 \left(\left\lceil \max_{v \in Vars}(|\text{log-likelihood}(\underline{v})|, |\text{log-likelihood}(\overline{v})|) \right\rceil \right)$$

$$F_{LL} = 32 - 1 - I_{LL}$$

Step 2) MCMC Code Instrumentation & Optimization

Dynamic Checks

- Overflows need to be checked for at runtime

Dynamic Checks

- Overflows need to be checked for at runtime
- Luckily, we don't have to check ***every*** arithmetic operation

Dynamic Checks

- Overflows need to be checked for at runtime
- Luckily, we don't have to check ***every*** arithmetic operation
- Only check ***final*** Acceptance Ratio summation

Domain Specific Optimizations

- MCMC sampling benefits from:

Domain Specific Optimizations

- MCMC sampling benefits from:

- Constant Propagation through the Bayesian Network

Domain Specific Optimizations

- MCMC sampling benefits from:

- Constant Propagation through the Bayesian Network

- Memoization during likelihood computation

How does Statheros perform?

Evaluation - Methodology

- Took multiple benchmarks from the Literature
- Run MCMC for 10K samples + 5k burn-in to get posterior
- Accuracy: $\left| \frac{\text{True Param Value - Posterior Mean}}{\text{True Param Value}} \right|$

Evaluation - Methodology

- Measure inference runtime and accuracy for Fixed Point (Statheros) against Float (32-bit) and Double (64-bit)

- Evaluated on 3 devices: Arduino (no FPU), Raspberry Pi and PocketBeagle

Evaluation - Methodology

- Measure inference runtime and accuracy for Fixed Point (Statheros) against Float (32-bit) and Double (64-bit)

- Evaluated on 3 devices: Arduino (no FPU), Raspberry Pi and PocketBeagle

ARM Cortex-M3 84 MHz

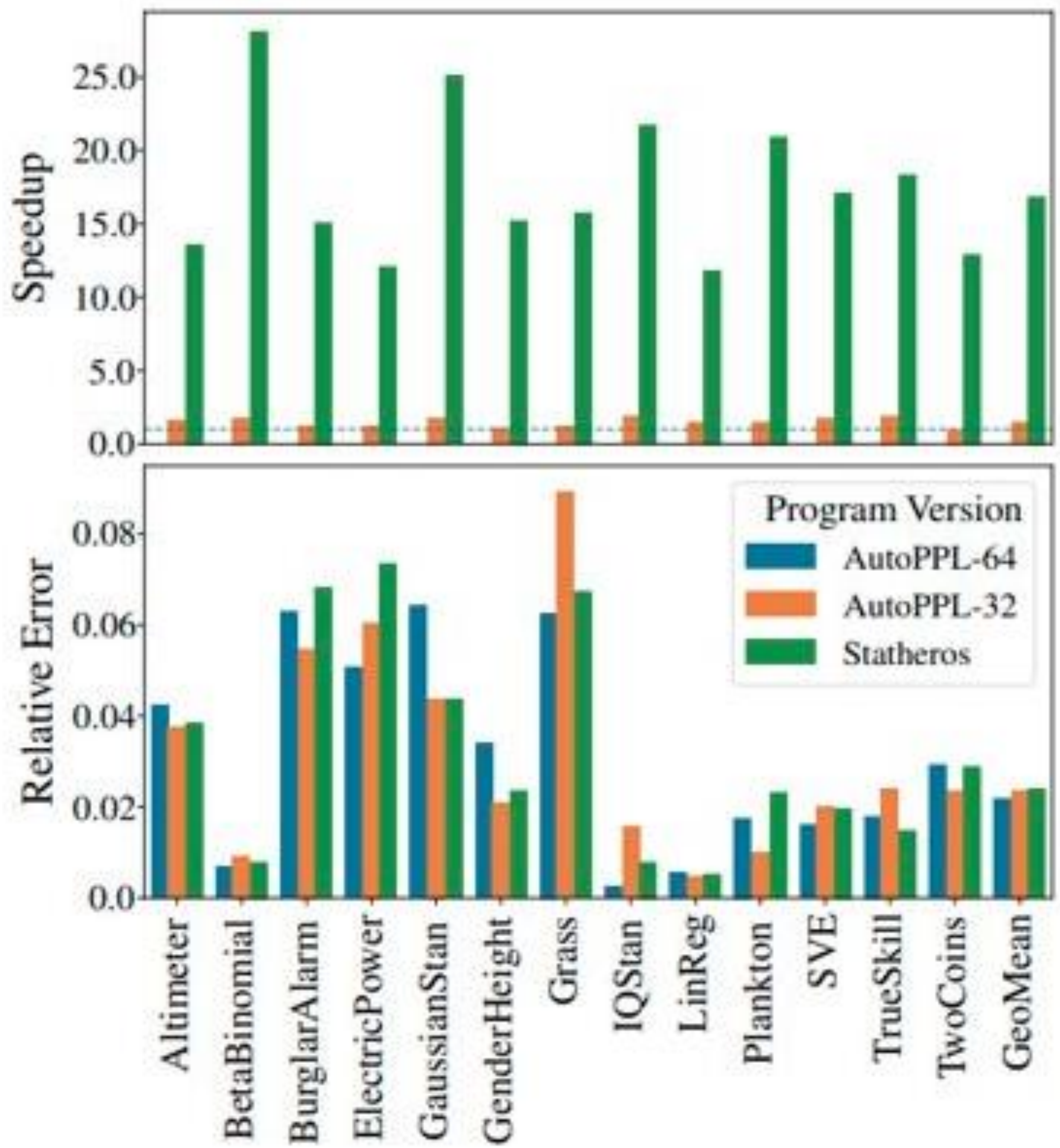
ARM Cortex-A53 1.4 GHz

ARM Cortex-A8 1 GHz

Benchmark	Distributions
Altimeter	Bernoulli
Beta-Binomial	Beta, Binomial
Burglar Alarm	Bernoulli
Electric Power	Bernoulli
Gaussian Stan	Gaussian
Gender Height	Bernoulli, Gaussian
Grass	Bernoulli
IQ Stan	Uniform, Gaussian
Linear Regression	Uniform, Gaussian
Plankton	Uniform, Gaussian
SVE	Uniform, Triangular, Gaussian
TrueSkill	Bernoulli, Gaussian
TwoCoins	Bernoulli

Arduino

- Substantial speedup due to Arduino's lack of FPU
- Statheros GeoMean Speedup:
  16.91x (over 64 bit double)
  11.54x (over 32 bit float)
- Geomean Relative Error
  Statheros: 0.0239
  32 bit float: 0.0238
  64 bit float: 0.0218

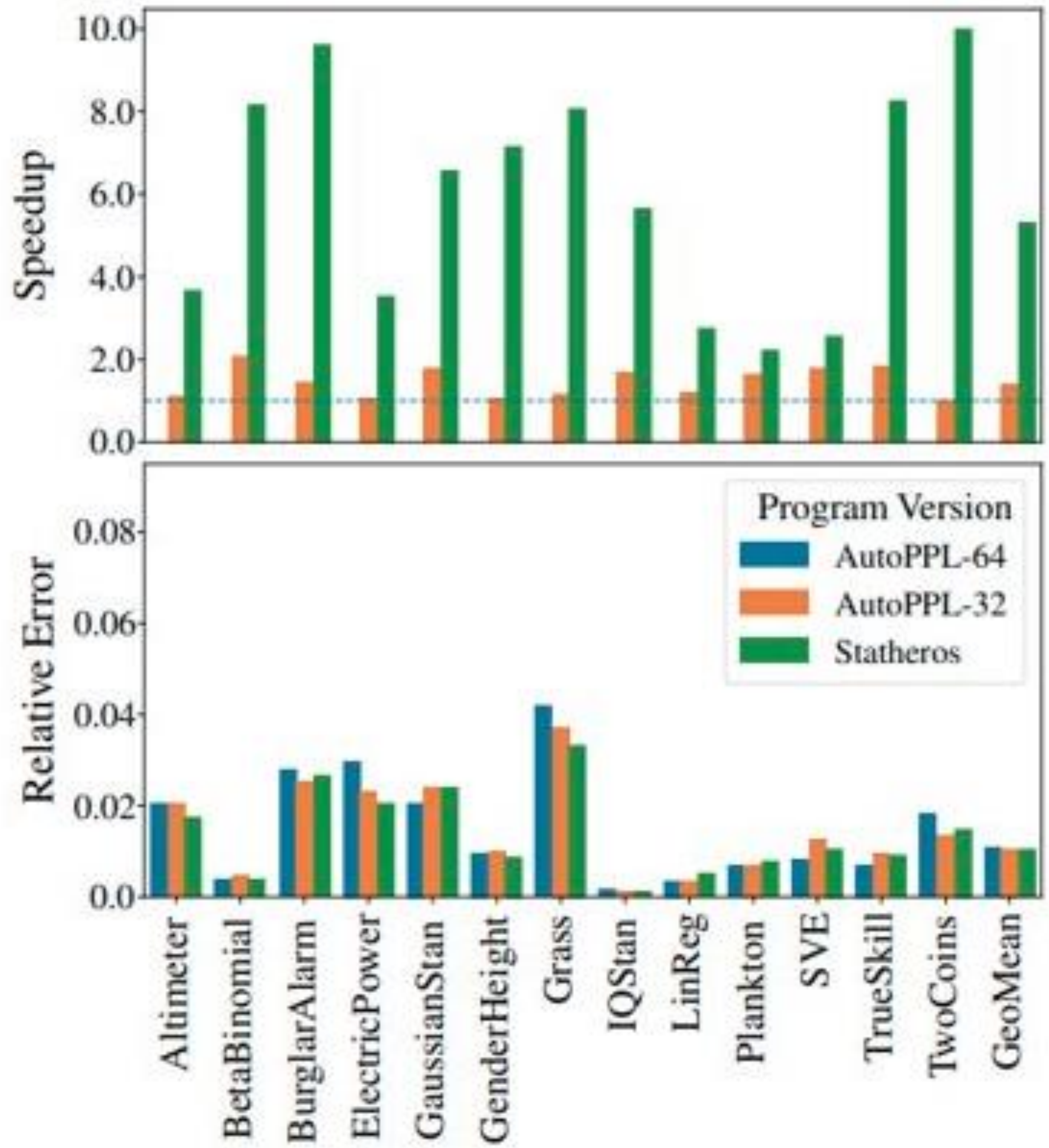


PocketBeagle

- PocketBeagle has low-end FPU -> still a large speedup

- Statheros GeoMean Speedup:
  5.33x (over 64 bit double)
  3.77x (over 32 bit float)

- Geomean Relative Error
  Statheros: 0.01
  32 bit float: 0.01
  64 bit float: 0.01

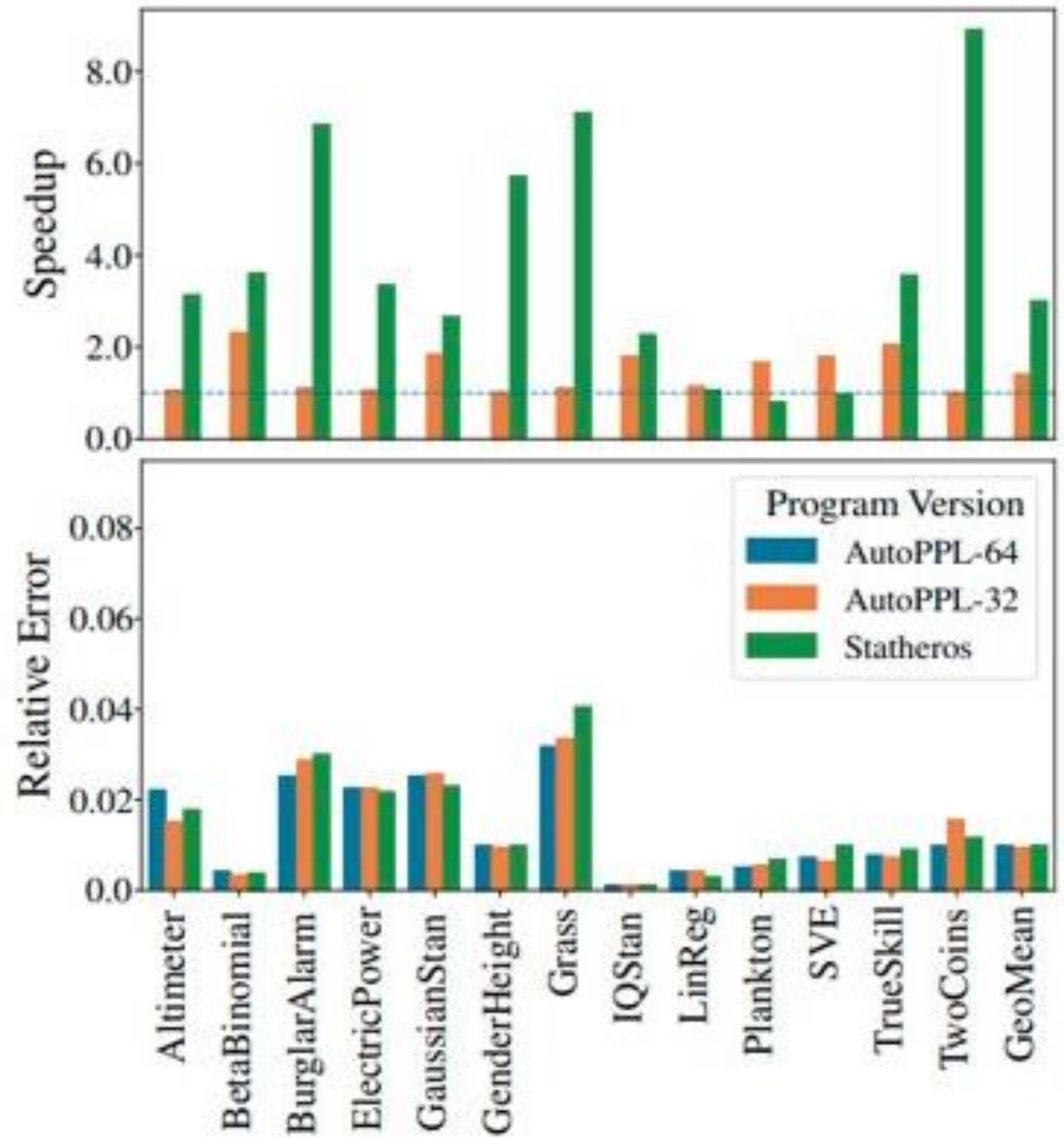


Raspberry Pi

- Raspberry Pi does have an FPU: speedup not as large

- Statheros GeoMean Speedup:
  3.04x (over 64 bit double)
  2.15x (over 32 bit float)

- Geomean Relative Error
  Statheros: 0.01
  32 bit float: 0.01
  64 bit float: 0.01



Benchmark	Parameter Configuration		Likelihood Configuration	
	Integer	Fractional	Integer	Fractional
Altimeter	7	24	7	24
Beta-Binomial	7	24	19	12
Burglar Alarm	7	24	7	24
Electric Power	7	24	7	24
Gaussian Stan	11	20	19	12
Gender Height	11	20	11	20
Grass	7	24	7	24
IQ Stan	11	20	19	12
Linear Regression	7	24	19	12
Plankton	7	24	7	24
SVE	7	24	19	12
TrueSkill	11	20	7	24
TwoCoins	7	24	7	24

Evaluation - Takeaways

- Statheros faster than float and double on all benchmarks for Arduino and PocketBeagle
- Inferred Fixed Point configurations tolerate approximation
- Not all overflows are bad!

More in the Paper:

- Detailed Algorithmic Description of Compilation
- Impact of Optimizations
- Discussion of Related Work

Statheros Takeaways

- Probabilistic Programming + Fixed Point Precision offers

Statheros Takeaways

- Probabilistic Programming + Fixed Point Precision offers

Major Runtime Savings!

Statheros Takeaways

- Probabilistic Programming + Fixed Point Precision offers

Major Runtime Savings!

With Small Quantization Error!