Symbolic disintegration with a variety of base measures

Praveen Narayanan
Chung-chieh Shan

~~October 2018~~
December 2019

Exact calculation

for approximate computation

Exact calculation
for approximate computation

Automatic differentiation
for gradient descent

Automatic disintegration
for inference and sampling

Exact calculation
for approximate computation

Automatic differentiation
for gradient descent

Automatic disintegration
for inference and sampling

A variety of base measures

$$(t,p) \leftsquigarrow \mu \quad = \quad \begin{array}{l} t \leftsquigarrow \beta \\ p \leftsquigarrow \kappa(t) \end{array} \quad : \mathbb{M}\,(T \times P)$$

Disintegration relates a joint measure $\mu : \mathbb{M}(T \times P)$

$$(t,p) \leftsquigarrow \mu \quad = \quad \begin{array}{l} t \leftsquigarrow \beta \\ p \leftsquigarrow \kappa(t) \end{array} \quad : \mathbb{M}(T \times P)$$

Disintegration relates a joint measure $\mu : \mathbb{M}\,(T \times P)$ with a base measure $\beta : \mathbb{M}\,T$

$$
\begin{aligned}
(t,p) \leftsquigarrow \mu \quad = \quad & t \leftsquigarrow \beta \\
& p \leftsquigarrow \kappa(t)
\end{aligned}
\qquad : \mathbb{M}\,(T \times P)
$$

Disintegration relates a joint measure $\mu : \mathbb{M}\,(T \times P)$ with a base measure $\beta : \mathbb{M}\,T$ and a kernel $\kappa : T \to \mathbb{M}\,P$

by the equation

$$(t, p) \leftsquigarrow \mu \quad = \quad \begin{array}{l} t \leftsquigarrow \beta \\ p \leftsquigarrow \kappa(t) \end{array} \quad : \mathbb{M}\,(T \times P)$$

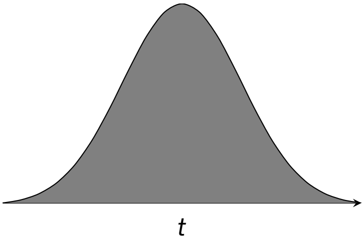
Disintegration relates **a joint measure** $\mu : \mathbb{M}\,(T \times P)$ with **a base measure** $\beta : \mathbb{M}\,T$ and **a kernel** $\kappa : T \to \mathbb{M}\,P$

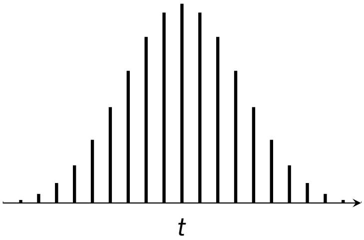
by the equation

$$(t,p) \leftsquigarrow \mu \quad = \quad \begin{array}{l} t \leftsquigarrow \beta \\ p \leftsquigarrow \kappa(t) \end{array} \quad : \mathbb{M}\,(T \times P)$$

Generalizes density:

$$\begin{array}{l} t \leftsquigarrow \textbf{normal} \\ p = () \end{array} \quad = \quad \begin{array}{l} t \leftsquigarrow \textbf{lebesgue} \\ p \leftsquigarrow \textbf{factor}\ (\textbf{dnorm}\ t) \end{array} \quad : \mathbb{M}\,(\mathbb{R} \times \mathbb{1})$$





Disintegration relates a joint measure $\mu : \mathbb{M}(T \times P)$ with a base measure $\beta : \mathbb{M}\,T$ and a kernel $\kappa : T \to \mathbb{M}\,P$

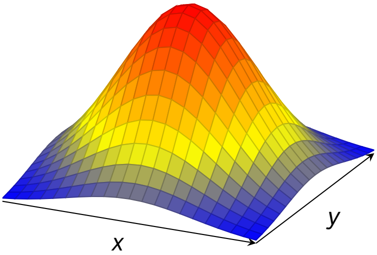
by the equation

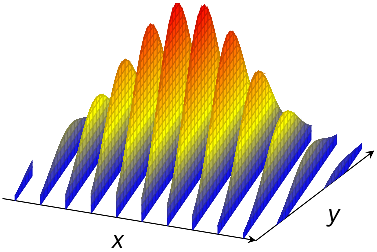
$$(t, p) \leftarrow \mu \quad = \quad \begin{array}{l} t \leftarrow \beta \\ p \leftarrow \kappa(t) \end{array} \quad : \mathbb{M}(T \times P)$$

Generalizes conditioning:

$$\begin{array}{l} x \leftarrow \textbf{normal} \\ y \leftarrow \textbf{normal} \\ t \leftarrow 5 \cdot x + 0.1 \cdot y \\ p = (x, y) \end{array} \quad = \quad \begin{array}{l} t \leftarrow \textbf{lebesgue} \\ p \leftarrow \ldots \Pr(p, t) \ldots \end{array} \quad : \mathbb{M}(\mathbb{R} \times \mathbb{R}^2)$$

"unnormalized conditioning"





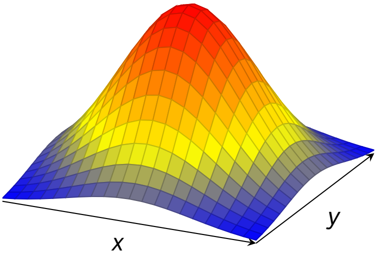
Disintegration relates **a joint measure** $\mu : \mathbb{M}\,(T \times P)$ with **a base measure** $\beta : \mathbb{M}\,T$ and **a kernel** $\kappa : T \to \mathbb{M}\,P$

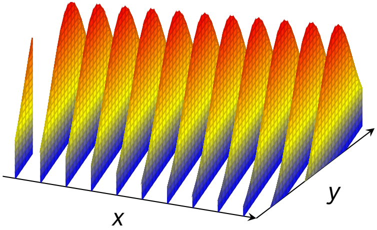
by the equation

$$(t, p) \leftsquigarrow \mu \quad = \quad \begin{array}{l} t \leftsquigarrow \beta \\ p \leftsquigarrow \kappa(t) \end{array} \quad : \mathbb{M}\,(T \times P)$$

Generalizes conditioning:

$$\begin{array}{l} x \leftsquigarrow \textbf{normal} \\ y \leftsquigarrow \textbf{normal} \\ t \leftsquigarrow 5 \cdot x + 0.1 \cdot y \\ p = (x, y) \end{array} \quad = \quad \begin{array}{l} t \leftsquigarrow \ldots \Pr(t) \ldots \\ p \leftsquigarrow \ldots \Pr(p|t) \ldots \end{array} \quad : \mathbb{M}\,(\mathbb{R} \times \mathbb{R}^2)$$





Disintegration relates **a joint measure** $\mu : \mathbb{M}(T \times P)$ with **a base measure** $\beta : \mathbb{M}\,T$ and **a kernel** $\kappa : T \to \mathbb{M}\,P$

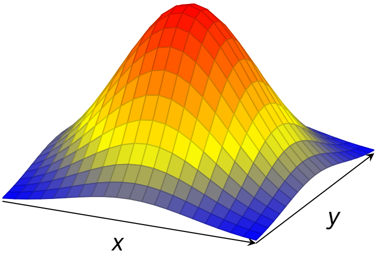
by the equation

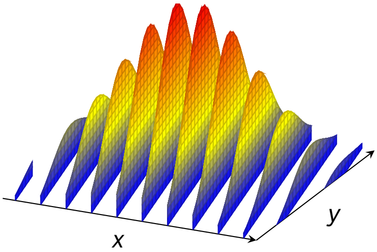
$$(t,p) \leftarrow\!\sim \mu \quad = \quad \begin{array}{l} t \leftarrow\!\sim \beta \\ p \leftarrow\!\sim \kappa(t) \end{array} \quad : \mathbb{M}(T \times P)$$

Generalizes conditioning:

$$\begin{array}{l} x \leftarrow\!\sim \textbf{normal} \\ y \leftarrow\!\sim \textbf{normal} \\ t \leftarrow\!\sim 5 \cdot x + 0.1 \cdot y \\ p = (x,y) \end{array} \quad = \quad \begin{array}{l} t \leftarrow\!\sim \textbf{lebesgue} \\ p \leftarrow\!\sim \ldots \Pr(p,t) \ldots \end{array} \quad : \mathbb{M}(\mathbb{R} \times \mathbb{R}^2)$$

"unnormalized conditioning"





Disintegration relates a joint measure $\mu : \mathbb{M}\,(T \times P)$ with a base measure $\beta : \mathbb{M}\,T$ and a kernel $\kappa : T \to \mathbb{M}\,P$

by the equation

$$(t, p) \leftsquigarrow \mu \quad = \quad \begin{array}{l} t \leftsquigarrow \beta \\ p \leftsquigarrow \kappa(t) \end{array} \quad : \mathbb{M}\,(T \times P)$$

Generalizes density and conditioning

Can be thought of as unnormalized conditioning

Disintegration relates a joint measure $\mu : \mathbb{M}\,(T \times P)$ with a base measure $\beta : \mathbb{M}\,T$ and a kernel $\kappa : T \to \mathbb{M}\,P$

by the equation

$$(t,p) \leftsquigarrow \mu \quad = \quad \begin{array}{l} t \leftsquigarrow \beta \\ p \leftsquigarrow \kappa(t) \end{array} \quad : \mathbb{M}\,(T \times P)$$

Generalizes density and conditioning

Can be thought of as unnormalized conditioning

A semantics-preserving transformation on probabilistic programs

Disintegration relates **a joint measure** $\mu : \mathbb{M}(T \times P)$ with **a base measure** $\beta : \mathbb{M}\,T$ and **a kernel** $\kappa : T \to \mathbb{M}\,P$

by the equation

$$(t,p) \leftarrow\!\!\sim \mu \;=\; \begin{array}{l} t \leftarrow\!\!\sim \beta \\ p \leftarrow\!\!\sim \kappa(t) \end{array} \;:\; \mathbb{M}(T \times P)$$

Generalizes density and conditioning

Can be thought of as unnormalized conditioning

A semantics-preserving transformation on probabilistic programs

random choice (**normal**), scoring (**factor**)

Disintegration relates a joint measure $\mu : \mathbb{M}(T \times P)$ with a base measure $\beta : \mathbb{M}\, T$ and a kernel $\kappa : T \to \mathbb{M}\, P$

by the equation

$$\begin{array}{l} (t,p) \leftarrow \mu \end{array} \quad = \quad \begin{array}{l} t \leftarrow \beta \\ p \leftarrow \kappa(t) \end{array} \quad : \mathbb{M}(T \times P)$$

Generalizes density and conditioning

Can be thought of as unnormalized conditioning

A semantics-preserving transformation on probabilistic programs

random choice (**normal**), scoring (**factor**)

sequence of operations, not just a primitive

Disintegration relates a joint measure $\mu : \mathbb{M}(T \times P)$ with a base measure $\beta : \mathbb{M}\,T$ and a kernel $\kappa : T \to \mathbb{M}\,P$

by the equation

$$(t,p) \leftsquigarrow \mu \quad = \quad \begin{array}{l} t \leftsquigarrow \beta \\ p \leftsquigarrow \kappa(t) \end{array} \quad : \mathbb{M}(T \times P)$$

Generalizes density and conditioning

Can be thought of as unnormalized conditioning

A semantics-preserving transformation on probabilistic programs

- s-finite measures/kernels
- random choice (**normal**), scoring (**factor**)
- sequence of operations, not just a primitive

Disintegration relates **a joint measure** $\mu : \mathbb{M}(T \times P)$ with **a base measure** $\beta : \mathbb{M}\,T$ and **a kernel** $\kappa : T \to \mathbb{M}\,P$

by the equation

$$(t,p) \leftarrow\!\!\sim \mu \quad = \quad \begin{array}{l} t \leftarrow\!\!\sim \beta \\ p \leftarrow\!\!\sim \kappa(t) \end{array} \quad : \mathbb{M}(T \times P)$$

Generalizes density and conditioning

Can be thought of as unnormalized conditioning

A semantics-preserving transformation on probabilistic programs

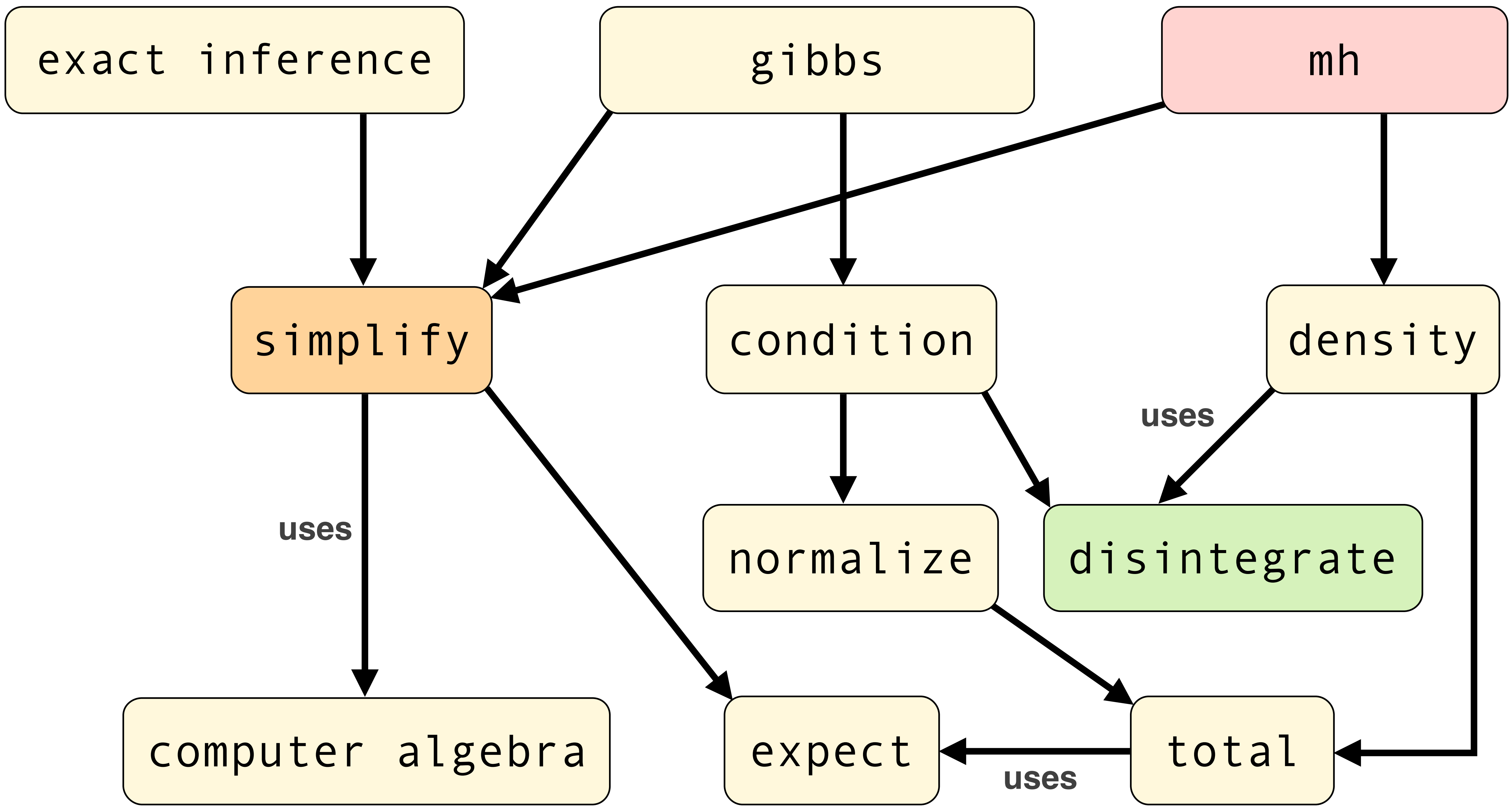
- s-finite measures/kernels
- random choice (**normal**), scoring (**factor**)
- sequence of operations, not just a primitive

Derived by equational reasoning (hence proven sound by construction)

= This talk: equational reasoning on example programs

\+ Established PL technology: lazy and partial evaluation (traversing computation graph)

Composable transformations

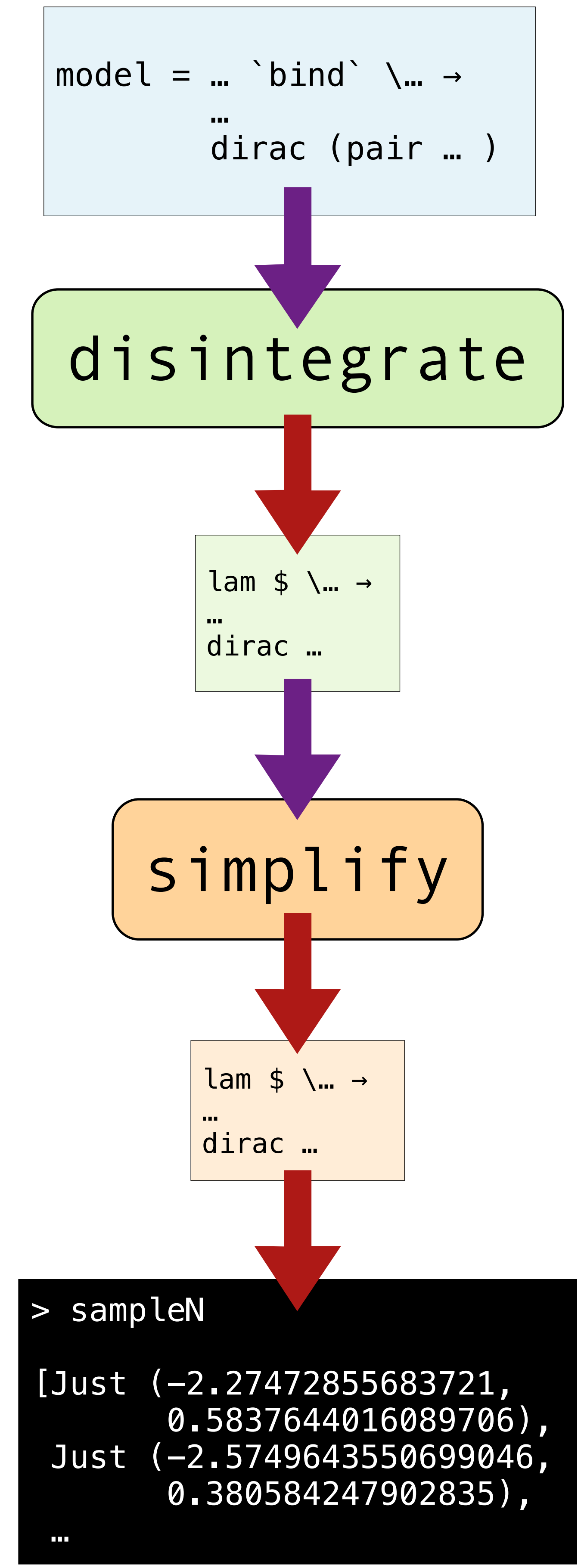


Modeling

Conditioning

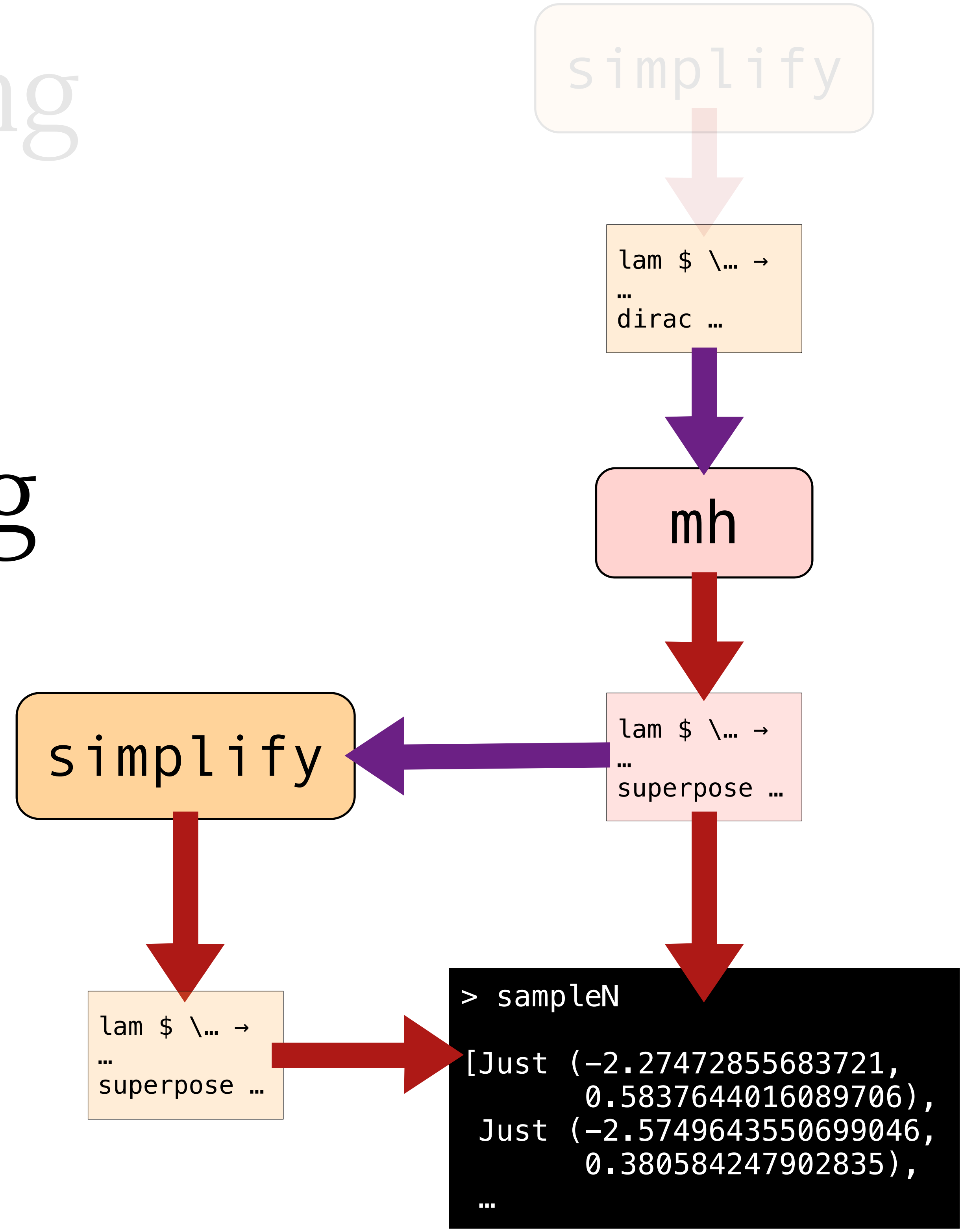
Optimizing

Sampling



Optimizing

Sampling



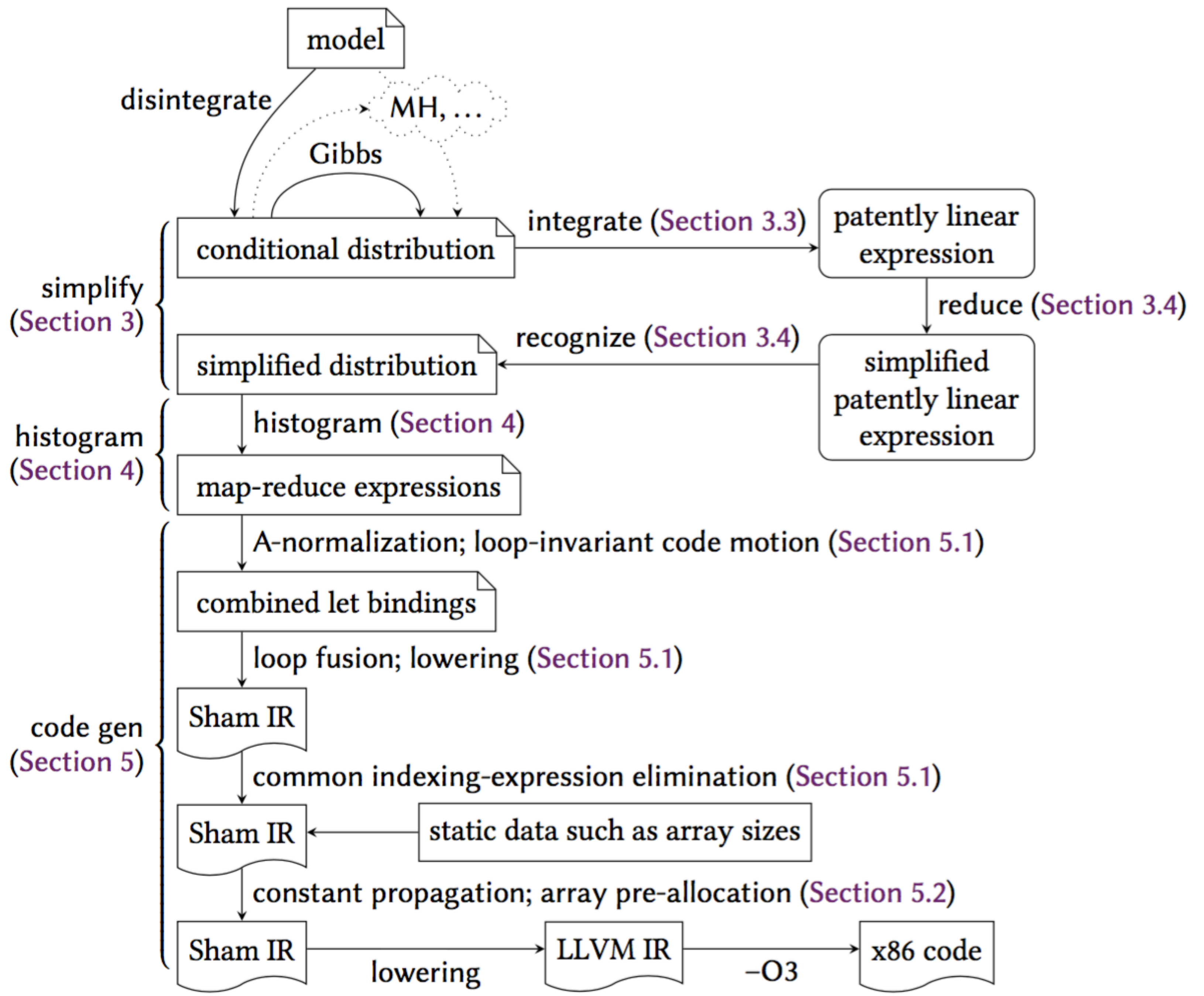


Fig. 1. Our pipeline, compiling probabilistic programs via math into imperative code to process data

A growing variety of base measures

Bhat et al. POPL 2012,
Shan & Ramsey POPL 2017

$$\beta ::= \textbf{lebesgue} \mid \textbf{counting} \mid \beta \otimes \beta$$

A growing variety of base measures

$$\beta ::= \overbrace{\beta \otimes \cdots \otimes \beta}^{n}$$

Narayanan & Shan
ICFP 2017

Bhat et al. POPL 2012,
Shan & Ramsey POPL 2017

$$\beta ::= \textbf{lebesgue} \mid \textbf{counting} \mid \beta \otimes \beta$$

A growing variety of base measures

$$\beta ::= \overbrace{\beta \otimes \cdots \otimes \beta}^{n}$$

Narayanan & Shan
ICFP 2017

Bhat et al. POPL 2012,
Shan & Ramsey POPL 2017

$$\beta ::= \textbf{lebesgue} \mid \textbf{counting} \mid \beta \otimes \beta$$

Narayanan & Shan

$$\beta ::= \textbf{ret}\ e \obar \beta \mid \beta \oplus \beta \mid (x \leftsquigarrow \beta) \otimes \beta$$

Discrete-continuous mixtures
Disjoint sums
Dependent products

A growing variety of base measures

$$\beta ::= \overbrace{\beta \otimes \cdots \otimes \beta}^{n}$$

Narayanan & Shan
ICFP 2017

Bhat et al. POPL 2012,
Shan & Ramsey POPL 2017

$$\beta ::= \textbf{lebesgue} \mid \textbf{counting} \mid \beta \otimes \beta$$

Narayanan & Shan

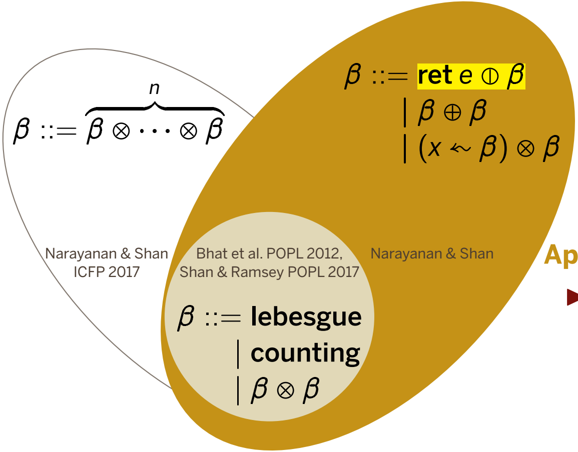
$$\beta ::= \textbf{ret}\ e \oplus \beta \mid \beta \oplus \beta \mid (x \leftsquigarrow \beta) \otimes \beta$$

Discrete-continuous mixtures
Disjoint sums
Dependent products

Applications:

- clamped model/observation
  (GPA, Tobit, camera)
  - likelihood ratio, importance sampling, mutual information
  - belief update, Gibbs sampling
- Metropolis-Hastings sampling
  - single site
  - reversible jump, light transport

A growing variety of base measures



Discrete-continuous mixtures
Disjoint sums
Dependent products

To appear in TOPLAS

Applications:

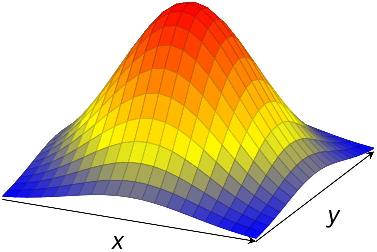
- clamped model/observation (GPA, Tobit, camera)
  - likelihood ratio, importance sampling, mutual information
  - belief update, Gibbs sampling
- Metropolis-Hastings sampling
  - single site
  - reversible jump, light transport

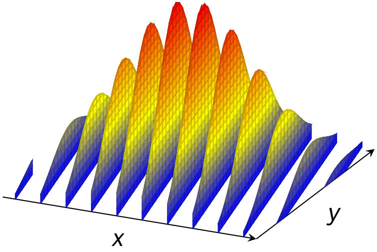
Fixed-base disintegration

$x \leftsquigarrow$ **normal**
$y \leftsquigarrow$ **normal**
$t = 5 \cdot x + 0.1 \cdot y$
$p = (x, y)$

$=$

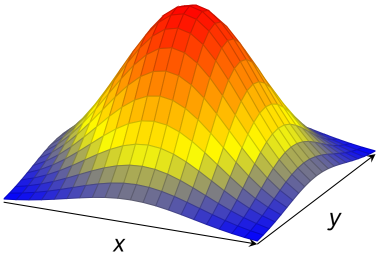
$t \leftsquigarrow$ **lebesgue**
$p \leftsquigarrow$ **???**

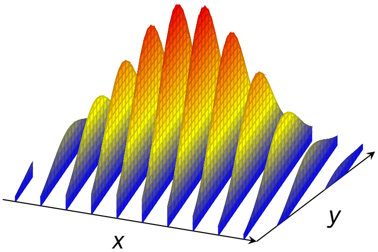




Fixed-base disintegration

$x \leftsquigarrow$ **normal** $y \leftsquigarrow$ **normal** $t = 5 \cdot x + 0.1 \cdot y$ $p = (x, y)$	$=$	$x \leftsquigarrow$ **normal** $y \leftsquigarrow$ **lebesgue** $t = 5 \cdot x + 0.1 \cdot y$ **factor** (**dnorm** y) $p = (x, y)$	$=$	$t \leftsquigarrow$ **lebesgue** $x \leftsquigarrow$ **normal** $y = 10 \cdot (t - 5 \cdot x)$ **factor** 10 **factor** (**dnorm** y) $p = (x, y)$





Fixed-base disintegration

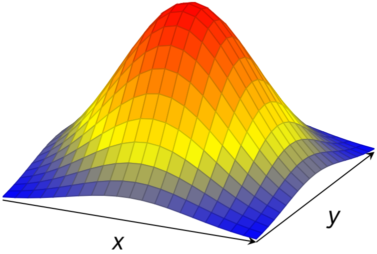
$x \leftsquigarrow$ **normal**
$y \leftsquigarrow$ **normal**
$t = 5 \cdot x + 0.1 \cdot y$
$p = (x, y)$

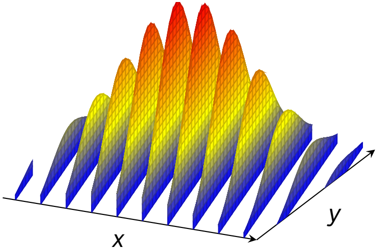
density

$x \leftsquigarrow$ **normal**
$y \leftsquigarrow$ **lebesgue**
$t = 5 \cdot x + 0.1 \cdot y$
factor (**dnorm** y)
$p = (x, y)$

reparameterize

$t \leftsquigarrow$ **lebesgue**
$x \leftsquigarrow$ **normal**
$y = 10 \cdot (t - 5 \cdot x)$
factor 10
factor (**dnorm** y)
$p = (x, y)$





Disintegration for *two* measurements

```
do {a ⇜ normal μ_a σ_a;
    b ⇜ normal μ_b σ_b;
    y_1 ⇜ normal (a · n_1 + b) 1;
    y_2 ⇜ normal (a · n_2 + b) 1;
    return ((y_1 , y_2) , (a , b))}
```

⇓

```
λ (t_1 , t_2) →
do {◁ (ȳ_1 , ȳ_2) (t_1 , t_2) [ā ⇜ normal μ_a σ_a;          ];
                              [b̄ ⇜ normal μ_b σ_b;          ]
                              [ȳ_1 ⇜ normal (a · n_1 + b) 1;]
                              [ȳ_2 ⇜ normal (a · n_2 + b) 1 ]
    return (a , b)}
```

⇓

```
λ (t_1 , t_2) →
do {◁ ȳ_1 t_1 [ā ⇜ normal μ_a σ_a;          ];
              [b̄ ⇜ normal μ_b σ_b;          ]
              [ȳ_1 ⇜ normal (a · n_1 + b) 1;]
              [ȳ_2 ⇜ normal (a · n_2 + b) 1 ]
    ◁ ȳ_2 t_2 ...;
    return (a , b)}
```

⇓

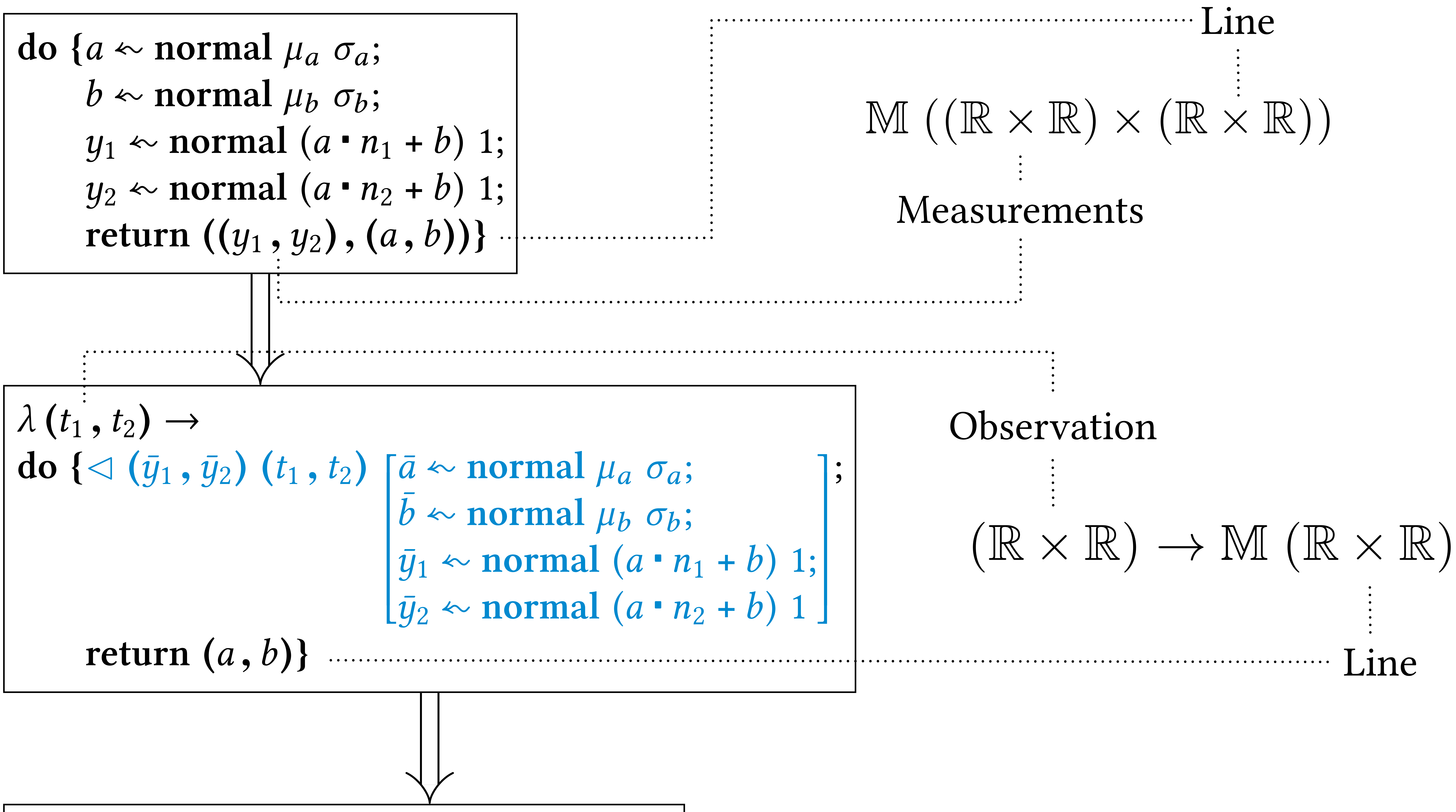
```
λ (t_1 , t_2) →
do {◁ ȳ_2 t_2 [ā ⇜ normal μ_a σ_a;                ];
              [b̄ ⇜ normal μ_b σ_b;                ]
              [factor (dNorm (a · n_1 + b) 1 t_1);]
              [ȳ_2 ⇜ normal (a · n_2 + b) 1       ]
    return (a , b)}
```

⇓

```
λ (t_1 , t_2) →
do {[ā ⇜ normal μ_a σ_a;                ];
    [b̄ ⇜ normal μ_b σ_b;                ]
    [factor (dNorm (a · n_1 + b) 1 t_1);]
    [factor (dNorm (a · n_2 + b) 1 t_2) ]
    return (a , b)}
```

⇓

```
λ (t_1 , t_2) →
do {a ⇜ normal μ_a σ_a;
    b ⇜ normal μ_b σ_b;
    factor (dNorm (a · n_1 + b) 1 t_1);
    factor (dNorm (a · n_2 + b) 1 t_2);
    return (a , b)}
```


do {a ⇜ normal μa σa;
b ⇜ normal μb σb;
y1 ⇜ normal (a · n1 + b) 1;
y2 ⇜ normal (a · n2 + b) 1;
return ((y1 , y2) , (a , b))}
Line
M ((ℝ × ℝ) × (ℝ × ℝ))
Measurements
λ (t1 , t2) →
do {◁ (ȳ1 , ȳ2) (t1 , t2) [ā ⇜ normal μa σa;
b̄ ⇜ normal μb σb;
ȳ1 ⇜ normal (a · n1 + b) 1;
ȳ2 ⇜ normal (a · n2 + b) 1];
return (a , b)}
Observation
(ℝ × ℝ) → M (ℝ × ℝ)
Line

$$
\begin{aligned}
&\mathbf{do}\ \{a \leftsquigarrow \mathbf{normal}\ \mu_a\ \sigma_a;\\
&\quad b \leftsquigarrow \mathbf{normal}\ \mu_b\ \sigma_b;\\
&\quad y_1 \leftsquigarrow \mathbf{normal}\ (a \cdot n_1 + b)\ 1;\\
&\quad y_2 \leftsquigarrow \mathbf{normal}\ (a \cdot n_2 + b)\ 1;\\
&\quad \mathbf{return}\ ((y_1\,, y_2)\,, (a\,, b))\}
\end{aligned}
$$

Constrain the measurements

to be the observation

$$
\begin{aligned}
&\lambda\,(t_1\,, t_2) \to\\
&\mathbf{do}\ \{\lhd\ (\bar{y}_1\,, \bar{y}_2)\ (t_1\,, t_2)\ \begin{bmatrix} \bar{a} \leftsquigarrow \mathbf{normal}\ \mu_a\ \sigma_a;\\ \bar{b} \leftsquigarrow \mathbf{normal}\ \mu_b\ \sigma_b;\\ \bar{y}_1 \leftsquigarrow \mathbf{normal}\ (a \cdot n_1 + b)\ 1;\\ \bar{y}_2 \leftsquigarrow \mathbf{normal}\ (a \cdot n_2 + b)\ 1 \end{bmatrix};\\
&\quad \mathbf{return}\ (a\,, b)\}
\end{aligned}
$$

in the context of the ***heap***

$\lambda\,(t_1\,,\,t_2) \rightarrow$

$\textbf{do}\ \{\triangleleft\ (\bar{y}_1\,,\,\bar{y}_2)\ (t_1\,,\,t_2)\ \begin{bmatrix} \bar{a} \leftsquigarrow \textbf{normal}\ \mu_a\ \sigma_a; \\ \bar{b} \leftsquigarrow \textbf{normal}\ \mu_b\ \sigma_b; \\ \bar{y}_1 \leftsquigarrow \textbf{normal}\ (a \cdot n_1 + b)\ 1; \\ \bar{y}_2 \leftsquigarrow \textbf{normal}\ (a \cdot n_2 + b)\ 1 \end{bmatrix};$

$\quad\textbf{return}\ (a\,,\,b)\}$

$\lambda\,(t_1\,,\,t_2) \rightarrow$

$\textbf{do}\ \{\triangleleft\ \bar{y}_1\ t_1\ \begin{bmatrix} \bar{a} \leftsquigarrow \textbf{normal}\ \mu_a\ \sigma_a; \\ \bar{b} \leftsquigarrow \textbf{normal}\ \mu_b\ \sigma_b; \\ \bar{y}_1 \leftsquigarrow \textbf{normal}\ (a \cdot n_1 + b)\ 1; \\ \bar{y}_2 \leftsquigarrow \textbf{normal}\ (a \cdot n_2 + b)\ 1 \end{bmatrix};$

$\quad\triangleleft\ \bar{y}_2\ t_2\ \ldots;$

$\quad\textbf{return}\ (a\,,\,b)\}$

Constrain is defined for each construct

$\triangleleft\ e\ t\ h =$

case e of

$(e + e') \rightarrow \ \ldots$

$(e\,,\,e') \rightarrow \ \ldots$

$\ldots$

Two recursive calls to ***constrain***

```
λ (t₁ , t₂) →
do {◁ ȳ₁ t₁ [ā ⇜ normal μ_a σ_a;
             b̄ ⇜ normal μ_b σ_b;
             ȳ₁ ⇜ normal (a · n₁ + b) 1;
             ȳ₂ ⇜ normal (a · n₂ + b) 1 ];
    ◁ ȳ₂ t₂ ...;
    return (a , b)}
```

Replace binding with normal density

```
λ (t₁ , t₂) →
do {◁ ȳ₂ t₂ [ā ⇜ normal μ_a σ_a;
             b̄ ⇜ normal μ_b σ_b;
             factor (dNorm (a · n₁ + b) 1 t₁);
             ȳ₂ ⇜ normal (a · n₂ + b) 1 ];
    return (a , b)}
```

$$
\begin{array}{l}
\lambda\,(t_1\,,\,t_2) \rightarrow \\
\textbf{do}\ \{\lhd\ \bar{y}_2\ t_2\ \left[\begin{array}{l}
\bar{a} \leftsquigarrow \textbf{normal}\ \mu_a\ \sigma_a; \\
\bar{b} \leftsquigarrow \textbf{normal}\ \mu_b\ \sigma_b; \\
\textbf{factor}\ (\textbf{dNorm}\ (a \cdot n_1 + b)\ 1\ t_1); \\
\bar{y}_2 \leftsquigarrow \textbf{normal}\ (a \cdot n_2 + b)\ 1
\end{array}\right]; \\
\quad\quad \textbf{return}\ (a\,,\,b)\}
\end{array}
$$

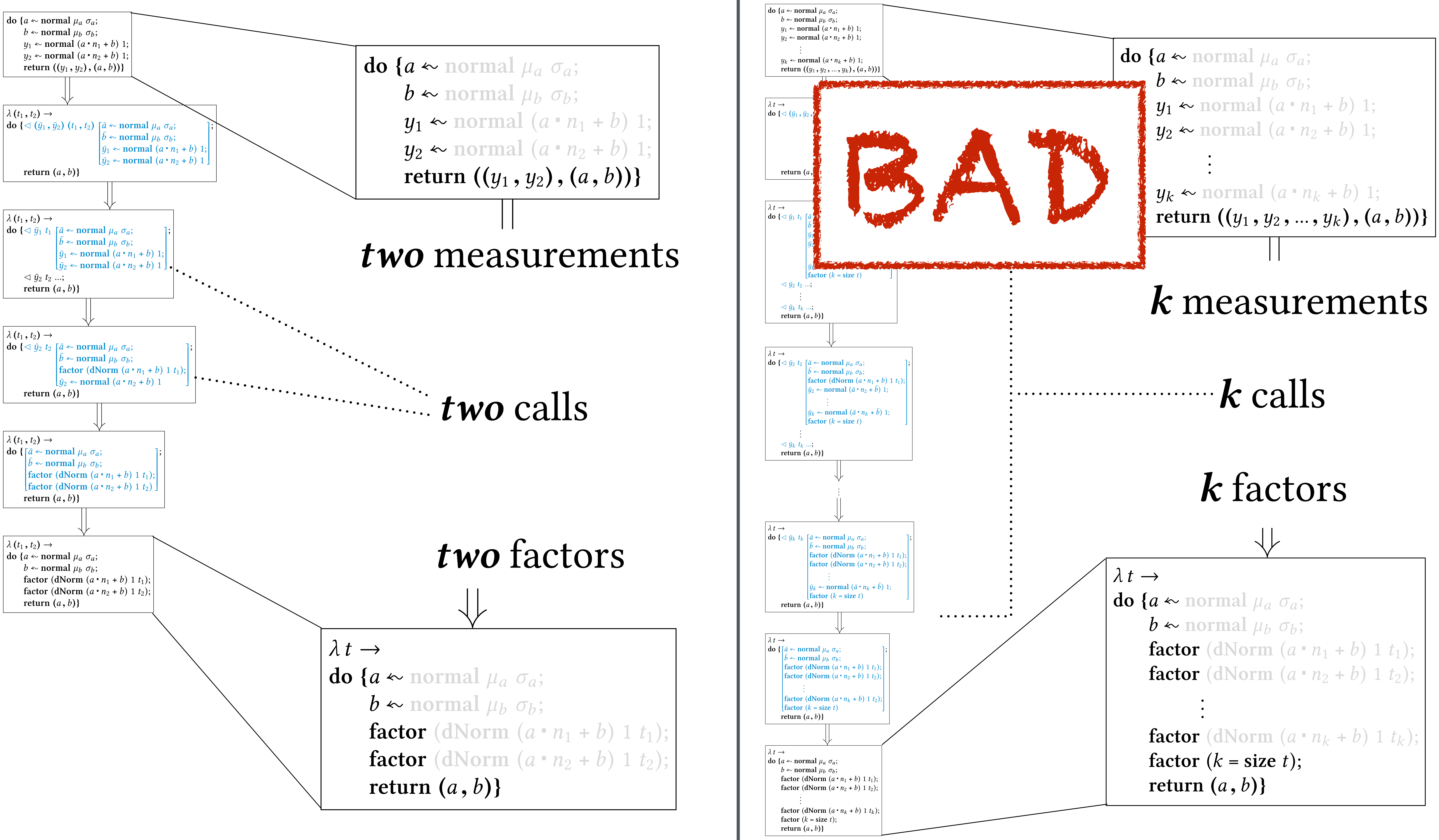
$$
\begin{array}{l}
\lambda\,(t_1\,,\,t_2) \rightarrow \\
\textbf{do}\ \{\left[\begin{array}{l}
\bar{a} \leftsquigarrow \textbf{normal}\ \mu_a\ \sigma_a; \\
\bar{b} \leftsquigarrow \textbf{normal}\ \mu_b\ \sigma_b; \\
\textbf{factor}\ (\textbf{dNorm}\ (a \cdot n_1 + b)\ 1\ t_1); \\
\textbf{factor}\ (\textbf{dNorm}\ (a \cdot n_2 + b)\ 1\ t_2)
\end{array}\right]; \\
\quad\quad \textbf{return}\ (a\,,\,b)\}
\end{array}
$$

Replace binding with normal density

$$
\begin{aligned}
&\lambda\,(t_1\,,\,t_2) \rightarrow\\
&\textbf{do}\ \{\left[\begin{array}{l}\bar{a} \leftsquigarrow \textbf{normal}\ \mu_a\ \sigma_a;\\ \bar{b} \leftsquigarrow \textbf{normal}\ \mu_b\ \sigma_b;\\ \textbf{factor}\ (\textbf{dNorm}\ (a \cdot n_1 + b)\ 1\ t_1);\\ \textbf{factor}\ (\textbf{dNorm}\ (a \cdot n_2 + b)\ 1\ t_2)\end{array}\right];\\
&\qquad \textbf{return}\ (a\,,\,b)\}
\end{aligned}
$$

Absorb heap
into
final output

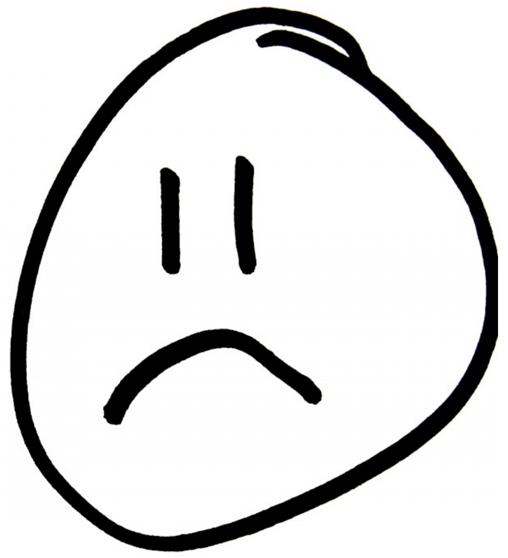
$$
\begin{aligned}
&\lambda\,(t_1\,,\,t_2) \rightarrow\\
&\textbf{do}\ \{a \leftsquigarrow \textbf{normal}\ \mu_a\ \sigma_a;\\
&\qquad b \leftsquigarrow \textbf{normal}\ \mu_b\ \sigma_b;\\
&\qquad \textbf{factor}\ (\textbf{dNorm}\ (a \cdot n_1 + b)\ 1\ t_1);\\
&\qquad \textbf{factor}\ (\textbf{dNorm}\ (a \cdot n_2 + b)\ 1\ t_2);\\
&\qquad \textbf{return}\ (a\,,\,b)\}
\end{aligned}
$$


do {a ↜ normal μa σa;
b ↜ normal μb σb;
y1 ↜ normal (a • n1 + b) 1;
y2 ↜ normal (a • n2 + b) 1;
return ((y1 , y2) , (a , b))}
two measurements
two calls
two factors
λ t →
do {a ↜ normal μa σa;
b ↜ normal μb σb;
factor (dNorm (a • n1 + b) 1 t1);
factor (dNorm (a • n2 + b) 1 t2);
return (a , b)}
BAD
do {a ↜ normal μa σa;
b ↜ normal μb σb;
y1 ↜ normal (a • n1 + b) 1;
y2 ↜ normal (a • n2 + b) 1;
yk ↜ normal (a • nk + b) 1;
return ((y1 , y2 , … , yk) , (a , b))}
k measurements
k calls
k factors
λ t →
do {a ↜ normal μa σa;
b ↜ normal μb σb;
factor (dNorm (a • n1 + b) 1 t1);
factor (dNorm (a • n2 + b) 1 t2);
factor (dNorm (a • nk + b) 1 tk);
factor (k = size t);
return (a , b)}

Unrolled

k calls

k factors



```
do {a ⇜ normal μ_a σ_a;
    b ⇜ normal μ_b σ_b;
    y_1 ⇜ normal (a • n_1 + b) 1;
    y_2 ⇜ normal (a • n_2 + b) 1;
        ⋮
    y_k ⇜ normal (a • n_k + b) 1;
    return ((y_1 , y_2 , … , y_k) , (a , b))}
```

```
λ t →
do {◁ (ȳ_1 , ȳ_2 , … , ȳ_k) t [ā ⇜ normal μ_a σ_a;
                               b̄ ⇜ normal μ_b σ_b;
                               ȳ_1 ⇜ normal (ā • n_1 + b̄) 1;
                               ȳ_2 ⇜ normal (ā • n_2 + b̄) 1;
                                   ⋮
                               ȳ_k ⇜ normal (ā • n_k + b̄) 1];
    return (a , b)}
```

```
λ t →
do {◁ ȳ_1 t_1 [ā ⇜ normal μ_a σ_a;
               b̄ ⇜ normal μ_b σ_b;
               ȳ_1 ⇜ normal (ā • n_1 + b̄) 1;
               ȳ_2 ⇜ normal (ā • n_2 + b̄) 1;
                   ⋮
               ȳ_k ⇜ normal (ā • n_k + b̄) 1;
               factor (k = size t)];
    ◁ ȳ_2 t_2 …;
        ⋮
    ◁ ȳ_k t_k …;
    return (a , b)}
```

```
λ t →
do {◁ ȳ_2 t_2 [ā ⇜ normal μ_a σ_a;
               b̄ ⇜ normal μ_b σ_b;
               factor (dNorm (a • n_1 + b) 1 t_1);
               ȳ_2 ⇜ normal (ā • n_2 + b̄) 1;
                   ⋮
               ȳ_k ⇜ normal (ā • n_k + b̄) 1;
               factor (k = size t)];
        ⋮
    ◁ ȳ_k t_k …;
    return (a , b)}
```

⋮

```
λ t →
do {◁ ȳ_k t_k [ā ⇜ normal μ_a σ_a;
               b̄ ⇜ normal μ_b σ_b;
               factor (dNorm (a • n_1 + b) 1 t_1);
               factor (dNorm (a • n_2 + b) 1 t_2);
                   ⋮
               ȳ_k ⇜ normal (ā • n_k + b̄) 1;
               factor (k = size t)];
    return (a , b)}
```

```
λ t →
do {[ā ⇜ normal μ_a σ_a;
     b̄ ⇜ normal μ_b σ_b;
     factor (dNorm (a • n_1 + b) 1 t_1);
     factor (dNorm (a • n_2 + b) 1 t_2);
         ⋮
     factor (dNorm (a • n_k + b) 1 t_2);
     factor (k = size t)];
    return (a , b)}
```

```
λ t →
do {a ⇜ normal μ_a σ_a;
    b ⇜ normal μ_b σ_b;
    factor (dNorm (a • n_1 + b) 1 t_1);
    factor (dNorm (a • n_2 + b) 1 t_2);
        ⋮
    factor (dNorm (a • n_k + b) 1 t_k);
    factor (k = size t);
    return (a , b)}
```

```
do {a ⇜ normal μ_a σ_a;
    b ⇜ normal μ_b σ_b;
    y ⇜ plate k (λ i → normal (a • n_i + b) 1);
    return (y , (a , b))}
```

```
λ t →
do {◁ ȳ t [ā ⇜ normal μ_a σ_a;
           b̄ ⇜ normal μ_b σ_b;
           ȳ ⇜ plate k (λ i → normal (a • n_i + b) 1)];
    return (a , b)}
```

```
λ t →
do {◁ [ĵ·k] ȳ[j] t_ĵ [ā ⇜ []. normal μ_a σ_a;
                      b̄ ⇜ []. normal μ_b σ_b;
                      ȳ ⇜ [î·k]. normal (ā[] • n_î + b̄[]) 1;
                      factor []. (k = size t)];
    return (a , b)}
```

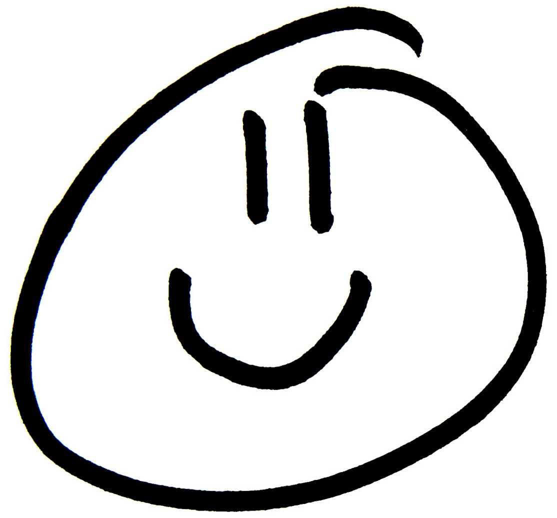
```
λ t →
do {[ā ⇜ []. normal μ_a σ_a;
     b̄ ⇜ []. normal μ_b σ_b;
     factor [ĵ·k]. (dNorm (ā[] • n_ĵ + b̄[]) 1 t_ĵ);
     factor []. (k = size t);
    return (a , b)}
```

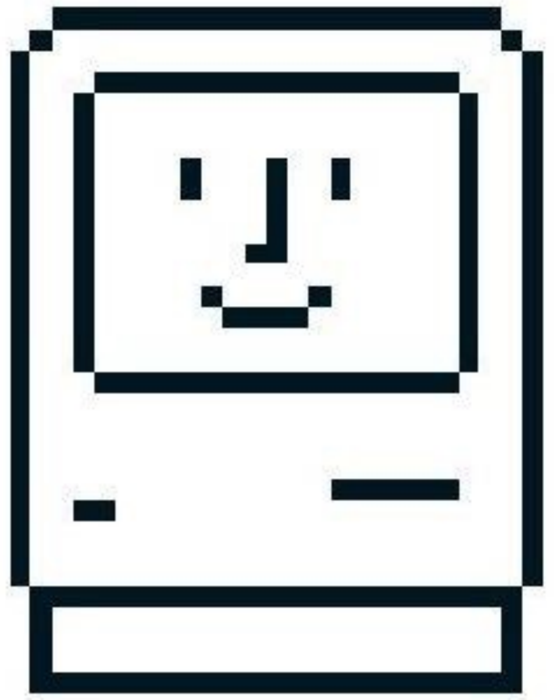
```
λ t →
do {a ⇜ normal μ_a σ_a;
    b ⇜ normal μ_b σ_b;
    _ ⇜ plate k (λ i → factor (dNorm (a • n_i + b) 1 t_i))
    factor (k = size t);
    return (a , b)}
```

Lifted

one call

one factor






Unrolled
Lifted
do {a ⇜ normal μa σa;
b ⇜ normal μb σb;
y1 ⇜ normal (a · n1 + b) 1;
y2 ⇜ normal (a · n2 + b) 1;
⋮
yk ⇜ normal (a · nk + b) 1;
return ((y1 , y2 , … , yk) , (a , b))}
become
do {a ⇜ normal μa σa;
b ⇜ normal μb σb;
y ⇜ plate k (λ i → normal (a · ni + b) 1);
return (y , (a , b))}
array
k-tuple
plate : ℕ → (ℕ → 𝕄 α) → 𝕄 ⟨α⟩
size
indexed dist.
dist. over arrays

Unrolled

$$
\begin{aligned}
&\textbf{do } \{a \leftsquigarrow \textbf{normal } \mu_a\ \sigma_a;\\
&\quad b \leftsquigarrow \textbf{normal } \mu_b\ \sigma_b;\\
&\quad y_1 \leftsquigarrow \textbf{normal } (a \cdot n_1 + b)\ 1;\\
&\quad y_2 \leftsquigarrow \textbf{normal } (a \cdot n_2 + b)\ 1;\\
&\qquad\qquad \vdots\\
&\quad y_k \leftsquigarrow \textbf{normal } (a \cdot n_k + b)\ 1;\\
&\quad \textbf{return } ((y_1, y_2, \ldots, y_k), (a, b))\}
\end{aligned}
$$

$$
\begin{aligned}
&\lambda\, t \rightarrow\\
&\textbf{do } \{\triangleleft\ (\bar{y}_1, \bar{y}_2, \ldots, \bar{y}_k)\ t\ \begin{bmatrix} \bar{a} \leftsquigarrow \textbf{normal } \mu_a\ \sigma_a;\\ \bar{b} \leftsquigarrow \textbf{normal } \mu_b\ \sigma_b;\\ \bar{y}_1 \leftsquigarrow \textbf{normal } (\bar{a} \cdot n_1 + \bar{b})\ 1;\\ \bar{y}_2 \leftsquigarrow \textbf{normal } (\bar{a} \cdot n_2 + \bar{b})\ 1;\\ \vdots\\ \bar{y}_k \leftsquigarrow \textbf{normal } (\bar{a} \cdot n_k + \bar{b})\ 1 \end{bmatrix};\\
&\quad \textbf{return } (a, b)\}
\end{aligned}
$$

Lifted

$$
\begin{aligned}
&\textbf{do } \{a \leftsquigarrow \textbf{normal } \mu_a\ \sigma_a;\\
&\quad b \leftsquigarrow \textbf{normal } \mu_b\ \sigma_b;\\
&\quad y \leftsquigarrow \textbf{plate } k\ (\lambda\, i \rightarrow \textbf{normal } (a \cdot n_i + b)\ 1);\\
&\quad \textbf{return } (y, (a, b))\}
\end{aligned}
$$

Unrolled

$$
\begin{aligned}
&\mathbf{do}\ \{a \leftsquigarrow \mathbf{normal}\ \mu_a\ \sigma_a;\\
&\quad b \leftsquigarrow \mathbf{normal}\ \mu_b\ \sigma_b;\\
&\quad y_1 \leftsquigarrow \mathbf{normal}\ (a \cdot n_1 + b)\ 1;\\
&\quad y_2 \leftsquigarrow \mathbf{normal}\ (a \cdot n_2 + b)\ 1;\\
&\qquad\quad \vdots\\
&\quad y_k \leftsquigarrow \mathbf{normal}\ (a \cdot n_k + b)\ 1;\\
&\quad \mathbf{return}\ ((y_1, y_2, \ldots, y_k), (a, b))\}
\end{aligned}
$$

$$\Downarrow$$

$$
\begin{aligned}
&\lambda\, t \rightarrow\\
&\mathbf{do}\ \{\triangleleft\ (\bar{y}_1, \bar{y}_2, \ldots, \bar{y}_k)\ t \begin{bmatrix} \bar{a} \leftsquigarrow \mathbf{normal}\ \mu_a\ \sigma_a;\\ \bar{b} \leftsquigarrow \mathbf{normal}\ \mu_b\ \sigma_b;\\ \bar{y}_1 \leftsquigarrow \mathbf{normal}\ (\bar{a} \cdot n_1 + \bar{b})\ 1;\\ \bar{y}_2 \leftsquigarrow \mathbf{normal}\ (\bar{a} \cdot n_2 + \bar{b})\ 1;\\ \vdots\\ \bar{y}_k \leftsquigarrow \mathbf{normal}\ (\bar{a} \cdot n_k + \bar{b})\ 1 \end{bmatrix};\\
&\quad \mathbf{return}\ (a, b)\}
\end{aligned}
$$

$$\Downarrow$$

$$\lambda\, t \rightarrow$$

Lifted

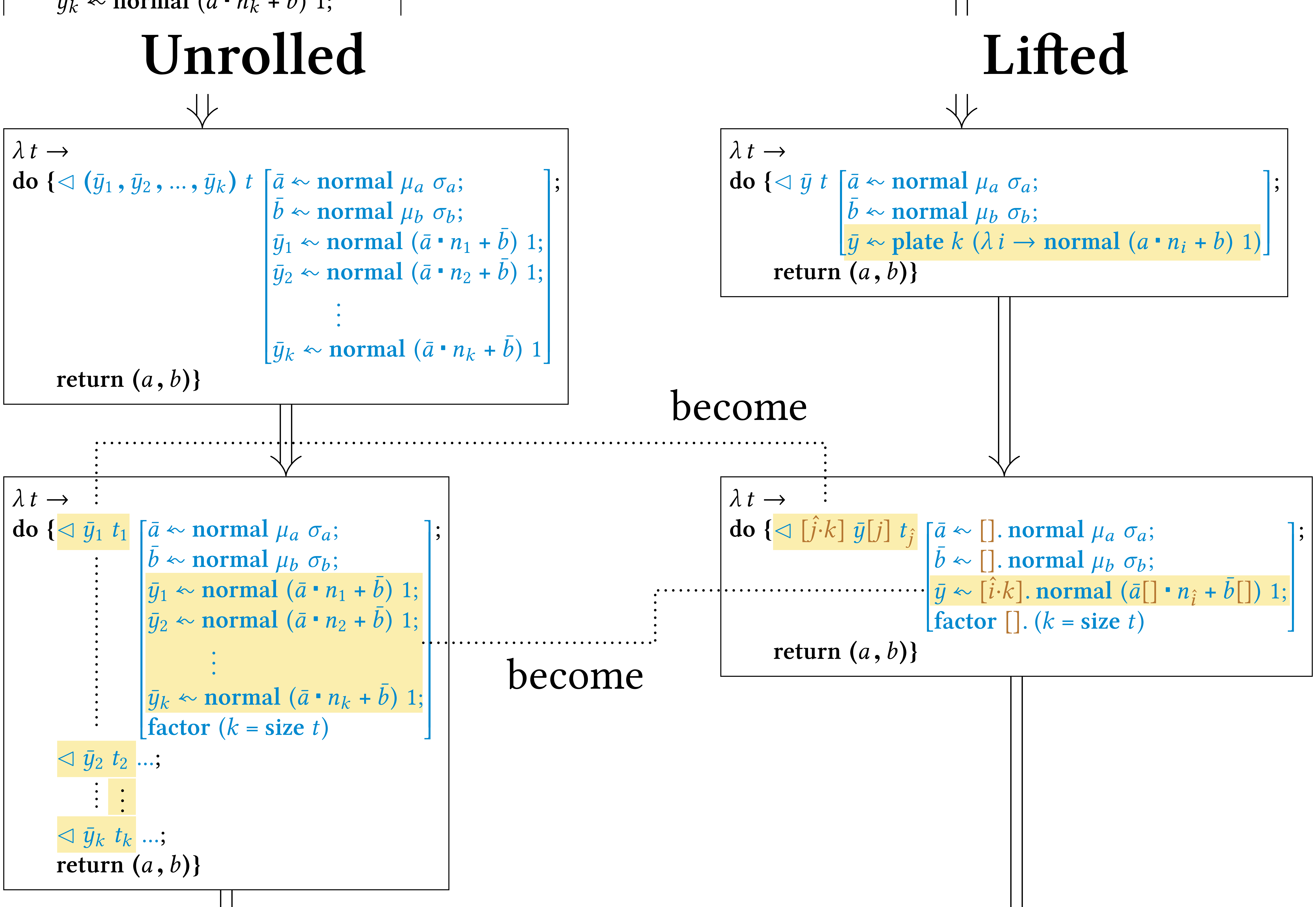
$$
\begin{aligned}
&\mathbf{do}\ \{a \leftsquigarrow \mathbf{normal}\ \mu_a\ \sigma_a;\\
&\quad b \leftsquigarrow \mathbf{normal}\ \mu_b\ \sigma_b;\\
&\quad y \leftsquigarrow \mathbf{plate}\ k\ (\lambda\, i \rightarrow \mathbf{normal}\ (a \cdot n_i + b)\ 1);\\
&\quad \mathbf{return}\ (y, (a, b))\}
\end{aligned}
$$

$$\Downarrow$$

$$
\begin{aligned}
&\lambda\, t \rightarrow\\
&\mathbf{do}\ \{\triangleleft\ \bar{y}\ t \begin{bmatrix} \bar{a} \leftsquigarrow \mathbf{normal}\ \mu_a\ \sigma_a;\\ \bar{b} \leftsquigarrow \mathbf{normal}\ \mu_b\ \sigma_b;\\ \bar{y} \leftsquigarrow \mathbf{plate}\ k\ (\lambda\, i \rightarrow \mathbf{normal}\ (a \cdot n_i + b)\ 1) \end{bmatrix};\\
&\quad \mathbf{return}\ (a, b)\}
\end{aligned}
$$

$$\Downarrow$$

$$\lambda\, t \rightarrow$$



Unrolled
Lifted
λ t →
do {◁ (ȳ1 , ȳ2 , … , ȳk) t [ā ⇜ normal μa σa;
b̄ ⇜ normal μb σb;
ȳ1 ⇜ normal (ā · n1 + b̄) 1;
ȳ2 ⇜ normal (ā · n2 + b̄) 1;
ȳk ⇜ normal (ā · nk + b̄) 1];
return (a , b)}
λ t →
do {◁ ȳ t [ā ⇜ normal μa σa;
b̄ ⇜ normal μb σb;
ȳ ⇜ plate k (λ i → normal (a · ni + b) 1)];
return (a , b)}
become
λ t →
do {◁ ȳ1 t1 [ā ⇜ normal μa σa;
b̄ ⇜ normal μb σb;
ȳ1 ⇜ normal (ā · n1 + b̄) 1;
ȳ2 ⇜ normal (ā · n2 + b̄) 1;
ȳk ⇜ normal (ā · nk + b̄) 1;
factor (k = size t)];
◁ ȳ2 t2 …;
◁ ȳk tk …;
return (a , b)}
become
λ t →
do {◁ [ĵ·k] ȳ[j] tĵ [ā ⇜ []. normal μa σa;
b̄ ⇜ []. normal μb σb;
ȳ ⇜ [î·k]. normal (ā[] · nî + b̄[]) 1;
factor []. (k = size t)];
return (a , b)}

Lifted

$$
\begin{array}{l}
\lambda\, t \to \\
\textbf{do}\ \{ \triangleleft\ [\hat{j}\cdot k]\ \bar{y}[j]\ t_{\hat{j}}\ \left[\begin{array}{l} \bar{a} \leftsquigarrow [].\ \textbf{normal}\ \mu_a\ \sigma_a; \\ \bar{b} \leftsquigarrow [].\ \textbf{normal}\ \mu_b\ \sigma_b; \\ \bar{y} \leftsquigarrow [\hat{i}\cdot k].\ \textbf{normal}\ (\bar{a}[] \cdot n_{\hat{i}} + \bar{b}[])\ 1; \\ \textbf{factor}\ [].\ (k = \textbf{size}\ t) \end{array} \right]; \\
\qquad \textbf{return}\ (a, b) \}
\end{array}
$$

Lifted constrain

Lifted binding

```
[ȳₖ ⇜ normal (ā · nₖ + b̄) 1]
```

Unrolled

```
λ t →
do {◁ ȳ₁ t₁ [ā ⇜ normal μₐ σₐ;
             b̄ ⇜ normal μ_b σ_b;
             ȳ₁ ⇜ normal (ā · n₁ + b̄) 1;
             ȳ₂ ⇜ normal (ā · n₂ + b̄) 1;
                      ⋮
             ȳₖ ⇜ normal (ā · nₖ + b̄) 1;
             factor (k = size t)                ];
    ◁ ȳ₂ t₂ ...;
          ⋮
    ◁ ȳₖ tₖ ...;
    return (a, b)}
```

```
λ t →
do {◁ ȳ₂ t₂ [ā ⇜ normal μₐ σₐ;
             b̄ ⇜ normal μ_b σ_b;
             factor (dNorm (a · n₁ + b) 1 t₁);
             ȳ₂ ⇜ normal (ā · n₂ + b̄) 1;
                      ⋮
             ȳₖ ⇜ normal (ā · nₖ + b̄) 1;
             factor (k = size t)                ];
          ⋮
    ◁ ȳₖ tₖ ...;
    return (a, b)}
```

Lifted

```
λ t →
do {◁ [ĵ·k] ȳ[j] t_ĵ [ā ⇜ []. normal μₐ σₐ;
                      b̄ ⇜ []. normal μ_b σ_b;
                      ȳ ⇜ [î·k]. normal (ā[] · n_î + b̄[]) 1;
                      factor []. (k = size t)                ];
    return (a, b)}
```

Unrolled

$\lambda\, t \rightarrow$
do {$\triangleleft\ \bar{y}_2\ t_2$ $\left[\begin{array}{l} \bar{a} \leftsquigarrow \textbf{normal}\ \mu_a\ \sigma_a; \\ \bar{b} \leftsquigarrow \textbf{normal}\ \mu_b\ \sigma_b; \\ \textbf{factor}\ (\textbf{dNorm}\ (a \cdot n_1 + b)\ 1\ t_1); \\ \bar{y}_2 \leftsquigarrow \textbf{normal}\ (\bar{a} \cdot n_2 + \bar{b})\ 1; \\ \quad\vdots \\ \bar{y}_k \leftsquigarrow \textbf{normal}\ (\bar{a} \cdot n_k + \bar{b})\ 1; \\ \textbf{factor}\ (k = \textbf{size}\ t) \end{array}\right]$;
$\vdots$
$\triangleleft\ \bar{y}_k\ t_k\ \ldots$;
return (a, b)}

$\vdots$

$\lambda\, t \rightarrow$
do {$\triangleleft\ \bar{y}_k\ t_k$ $\left[\begin{array}{l} \bar{a} \leftsquigarrow \textbf{normal}\ \mu_a\ \sigma_a; \\ \bar{b} \leftsquigarrow \textbf{normal}\ \mu_b\ \sigma_b; \\ \textbf{factor}\ (\textbf{dNorm}\ (a \cdot n_1 + b)\ 1\ t_1); \\ \textbf{factor}\ (\textbf{dNorm}\ (a \cdot n_2 + b)\ 1\ t_2); \\ \quad\vdots \\ \bar{y}_k \leftsquigarrow \textbf{normal}\ (\bar{a} \cdot n_k + \bar{b})\ 1; \\ \textbf{factor}\ (k = \textbf{size}\ t) \end{array}\right]$;
return (a, b)}

Lifted

Unrolled

$$\lambda\, t \to$$
$$\mathbf{do}\ \{\lhd\ \bar{y}_k\ t_k\ \left[\begin{array}{l} \bar{a} \leftsquigarrow \mathbf{normal}\ \mu_a\ \sigma_a; \\ \bar{b} \leftsquigarrow \mathbf{normal}\ \mu_b\ \sigma_b; \\ \text{factor}\ (\text{dNorm}\ (a \cdot n_1 + b)\ 1\ t_1); \\ \text{factor}\ (\text{dNorm}\ (a \cdot n_2 + b)\ 1\ t_2); \\ \quad\vdots \\ \bar{y}_k \leftsquigarrow \mathbf{normal}\ (\bar{a} \cdot n_k + \bar{b})\ 1; \\ \text{factor}\ (k = \text{size}\ t) \end{array}\right];$$
$$\mathbf{return}\ (a, b)\}$$

$$\lambda\, t \to$$
$$\mathbf{do}\ \{\left[\begin{array}{l} \bar{a} \leftsquigarrow \mathbf{normal}\ \mu_a\ \sigma_a; \\ \bar{b} \leftsquigarrow \mathbf{normal}\ \mu_b\ \sigma_b; \\ \text{factor}\ (\text{dNorm}\ (a \cdot n_1 + b)\ 1\ t_1); \\ \text{factor}\ (\text{dNorm}\ (a \cdot n_2 + b)\ 1\ t_2); \\ \quad\vdots \\ \text{factor}\ (\text{dNorm}\ (a \cdot n_k + b)\ 1\ t_2); \\ \text{factor}\ (k = \text{size}\ t) \end{array}\right];$$
$$\mathbf{return}\ (a, b)\}$$

become

Lifted

$$\lambda\, t \to$$
$$\mathbf{do}\ \{\left[\begin{array}{l} \bar{a} \leftsquigarrow [\,].\ \mathbf{normal}\ \mu_a\ \sigma_a; \\ \bar{b} \leftsquigarrow [\,].\ \mathbf{normal}\ \mu_b\ \sigma_b; \\ \text{factor}\ [\hat{j} \cdot k].\ (\text{dNorm}\ (\bar{a}[\,] \cdot n_{\hat{j}} + \bar{b}[\,])\ 1\ t_{\hat{j}}); \\ \text{factor}\ [\,].\ (k = \text{size}\ t); \end{array}\right];$$
$$\mathbf{return}\ (a, b)\}$$

[factor (k = size t)   ]

Unrolled

```
λ t →
do {[ā ↜ normal μ_a σ_a;
     b̄ ↜ normal μ_b σ_b;
     factor (dNorm (a · n_1 + b) 1 t_1);
     factor (dNorm (a · n_2 + b) 1 t_2);
              ⋮
     factor (dNorm (a · n_k + b) 1 t_2);
     factor (k = size t)                 ];
    return (a, b)}
```

```
λ t →
do {a ↜ normal μ_a σ_a;
    b ↜ normal μ_b σ_b;
    factor (dNorm (a · n_1 + b) 1 t_1);
    factor (dNorm (a · n_2 + b) 1 t_2);
             ⋮
    factor (dNorm (a · n_k + b) 1 t_k);
    factor (k = size t);
    return (a, b)}
```

Lifted

```
λ t →
do {[ā ↜ []. normal μ_a σ_a;
     b̄ ↜ []. normal μ_b σ_b;
     factor [ĵ·k]. (dNorm (ā[] · n_ĵ + b̄[]) 1 t_ĵ);
     factor []. (k = size t);                       ];
    return (a, b)}
```

```
λ t →
do {a ↜ normal μ_a σ_a;
    b ↜ normal μ_b σ_b;
    _ ↜ plate k (λ i → factor (dNorm (a · n_i + b) 1 t_i))
    factor (k = size t);
    return (a, b)}
```

$$naiveBayes \; : \mathbb{N} \to \mathbb{N} \to \mathbb{N} \to \mathbb{N} \to \mathbb{M}\,(\langle\langle\mathbb{N}\rangle\rangle \times \langle\mathbb{N}\rangle) \tag{6.38}$$

$$\begin{aligned} naiveBayes = \lambda sizeVocab.\, \lambda numLabels.\, \lambda numDocs.\, \lambda sizeEachDoc. \\ \mathbf{do}\ \{\beta \leftsquigarrow \mathbf{plate}\ numLabels\ \_.\ \mathbf{dirichlet}\ (\mathbf{array}\ sizeVocab\ \_.\ 1); \\ \theta \leftsquigarrow \mathbf{dirichlet}\ (\mathbf{array}\ numLabels\ \_.\ 1); \\ \zeta \leftsquigarrow \mathbf{plate}\ numDocs\ \_.\ \mathbf{categorical}\ \theta; \\ ds \leftsquigarrow \mathbf{plate}\ numDocs\ i. \\ \mathbf{plate}\ sizeEachDoc\ \_.\ \mathbf{categorical}\ \beta[\zeta[i]]; \\ \mathbf{return}\ (ds, \zeta)\} \end{aligned} \tag{6.39}$$

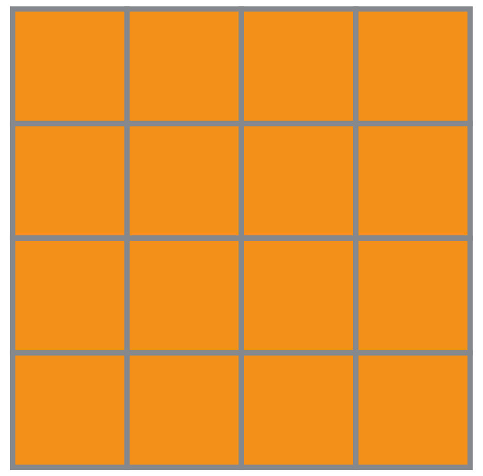
$$\mathit{disintegrate}^{\diamond}\ (\mathit{naiveBayes}\ \mathit{sizeVocab}\ \mathit{numLabels}\ \mathit{numDocs}\ \mathit{sizeEachDoc})\ t \tag{6.41}$$

$$\begin{aligned}
\Longrightarrow \mathbf{do}\ \{&\beta \leftsquigarrow \mathbf{plate}\ \mathit{numLabels}\ _.\ \mathbf{dirichlet}\ (\mathbf{array}\ \mathit{sizeVocab}\ _.\ 1); \\
&\theta \leftsquigarrow \mathbf{dirichlet}\ (\mathbf{array}\ \mathit{numLabels}\ _.\ 1); \\
&\zeta \leftsquigarrow \mathbf{plate}\ \mathit{numDocs}\ _.\ \mathbf{categorical}\ \theta; \\
&_ \leftsquigarrow \mathbf{plate}\ \mathit{numDocs}\ i. \\
&\qquad \mathbf{plate}\ \mathit{sizeEachDoc}\ j. \\
&\qquad\quad \mathbf{do}\ \{\mathbf{factor}\ (\beta[\zeta[i]][t[i][j]] \div \mathit{sum}\ \beta[\zeta[i]]); \\
&\qquad\qquad\quad \mathbf{return}\ ()\}; \\
&\mathbf{return}\ \zeta\}
\end{aligned} \tag{6.42}$$

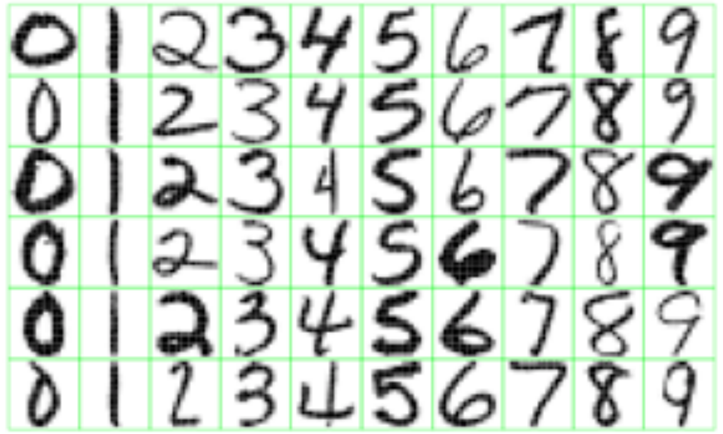
What works

$\hat{i} \cdot k$

Index variable — Size

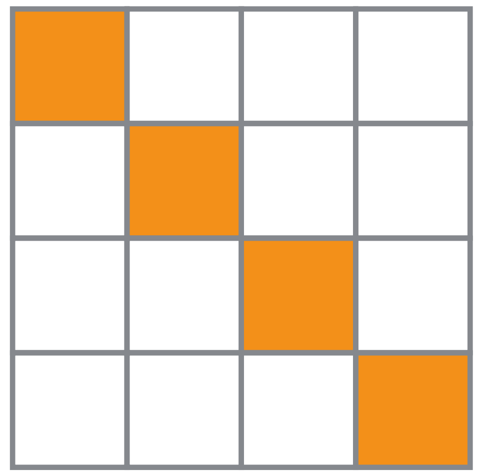


Observing whole arrays, with copying, mapping, and transposing

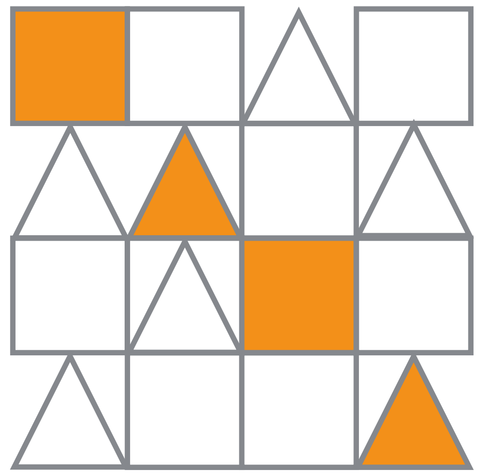


Models from R2 probabilistic programming suite

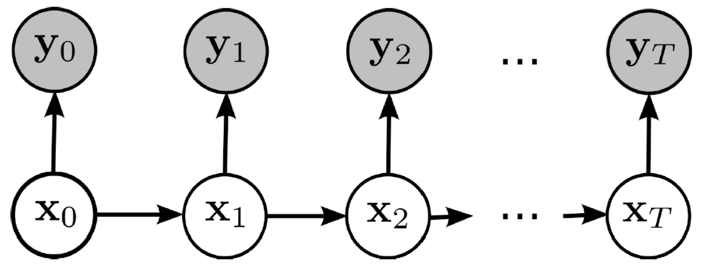
What we want to work



Observing parts of arrays

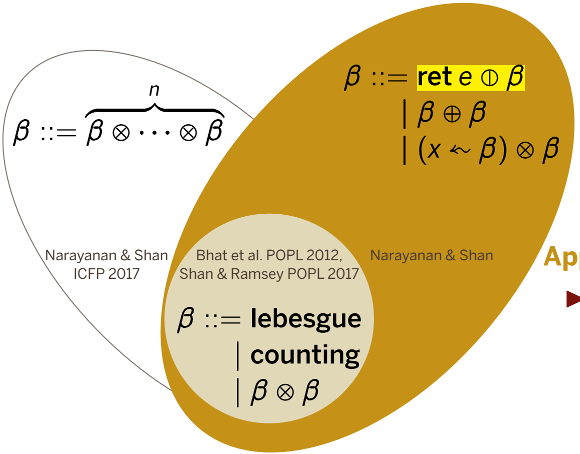


Conditionals inside arrays



Chain-like dependencies between array elements

A growing variety of base measures



Discrete-continuous mixtures
Disjoint sums
Dependent products

Applications:

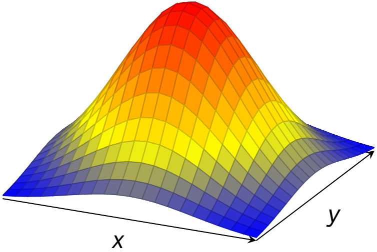
- clamped model/observation (GPA, Tobit, camera)
  - likelihood ratio, importance sampling, mutual information
  - belief update, Gibbs sampling
- Metropolis-Hastings sampling
  - single site
  - reversible jump, light transport

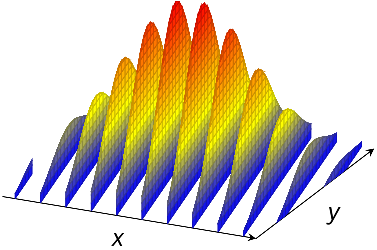
Clamped observation requires mixed base

$x \leftsquigarrow$ **normal**
$y \leftsquigarrow$ **normal**
$t = 5 \cdot x + 0.1 \cdot y$
$t' = \max\{-3, \min\{+3, t\}\}$
$p = (x, y)$

$\neq$

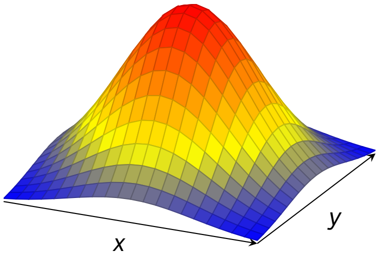
$t' \leftsquigarrow$ **lebesgue**
$p \leftsquigarrow$ **???**

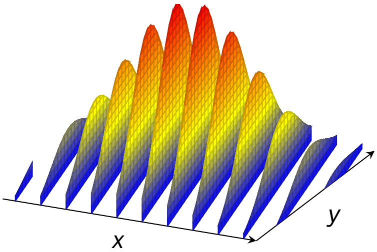




Clamped observation requires mixed base

$$
\begin{aligned}
x &\leftsquigarrow \textbf{normal} \\
y &\leftsquigarrow \textbf{normal} \\
t &= 5 \cdot x + 0.1 \cdot y \\
t' &\leftsquigarrow (\textbf{factor}\ \langle t < -3 \rangle ;\ \textbf{ret}\ {-3}) \oplus \\
&\quad\ (\textbf{factor}\ \langle t < +3 \rangle ;\ \textbf{ret}\ {+3}) \oplus \\
&\quad\ (\textbf{factor}\ \langle |t| \leq 3 \rangle ;\ \textbf{ret}\ t\ ) \\
p &= (x, y)
\end{aligned}
\quad = \quad
\begin{aligned}
t' &\leftsquigarrow \textbf{ret}\ {-3} \oplus \textbf{ret}\ {+3} \oplus \textbf{lebesgue} \\
p &\leftsquigarrow \textbf{???}
\end{aligned}
$$





Base-checking disintegration

$x \leftsquigarrow$ **normal**
$y \leftsquigarrow$ **normal**
$t = 5 \cdot x + 0.1 \cdot y$
factor $\langle t < -3 \rangle$
$t' = -3$
$p = (x, y)$

$=$

$t' \leftsquigarrow$ **ret** $-3 \oplus$ **ret** $+3 \oplus$ **lebesgue**
$p \leftsquigarrow$ **???**

$x \leftsquigarrow$ **normal**
$y \leftsquigarrow$ **normal**
$t = 5 \cdot x + 0.1 \cdot y$
factor $\langle |t| \leq 3 \rangle$
$p = (x, y)$

$=$

$t \leftsquigarrow$ **ret** $-3 \oplus$ **ret** $+3 \oplus$ **lebesgue**
$p \leftsquigarrow$ **???**

Base-checking disintegration

$x \leftsquigarrow$ **normal**
$y \leftsquigarrow$ **normal**
$t = 5 \cdot x + 0.1 \cdot y$
factor $\langle t < -3 \rangle$
$t' = -3$
$p = (x, y)$

density

$t' \leftsquigarrow$ **ret** $-3 \oplus$ **ret** $+3 \oplus$ **lebesgue**
factor $\langle t' = -3 \rangle$
$x \leftsquigarrow$ **normal**
$y \leftsquigarrow$ **normal**
$t = 5 \cdot x + 0.1 \cdot y$
factor $\langle t < -3 \rangle$
$p = (x, y)$

$x \leftsquigarrow$ **normal**
$y \leftsquigarrow$ **normal**
$t = 5 \cdot x + 0.1 \cdot y$
factor $\langle |t| \leq 3 \rangle$
$p = (x, y)$

density

$x \leftsquigarrow$ **normal**
$y \leftsquigarrow$ **ret** $(10 \cdot (-3 - 5 \cdot x)) \oplus$
ret $(10 \cdot (+3 - 5 \cdot x)) \oplus$
lebesgue
$t = 5 \cdot x + 0.1 \cdot y$
factor $\langle t \notin \{\pm 3\} \rangle$
factor $($**dnorm** $y)$
factor $\langle |t| \leq 3 \rangle$
$p = (x, y)$

reparameterize

$t \leftsquigarrow$ **ret** $-3 \oplus$ **ret** $+3 \oplus$ **lebesgue**
$x \leftsquigarrow$ **normal**
$y = 10 \cdot (t - 5 \cdot x)$
factor 10
factor $\langle t \notin \{\pm 3\} \rangle$
factor $($**dnorm** $y)$
factor $\langle |t| \leq 3 \rangle$
$p = (x, y)$

Base-inferring disintegration

$x \leftsquigarrow \textbf{normal}$ $y \leftsquigarrow \textbf{normal}$ $t = 5 \cdot x + 0.1 \cdot y$ $\textbf{factor}\ \langle t < -3 \rangle$ $t' = -3$ $p = (x, y)$	$=$	$t' \leftsquigarrow \mathbb{B}$ $p \leftsquigarrow \textbf{???}$
$x \leftsquigarrow \textbf{normal}$ $y \leftsquigarrow \textbf{normal}$ $t = 5 \cdot x + 0.1 \cdot y$ $\textbf{factor}\ \langle \lvert t \rvert \leq 3 \rangle$ $p = (x, y)$	$=$	$t \leftsquigarrow \mathbb{B}$ $p \leftsquigarrow \textbf{???}$

Base-inferring disintegration

$x \leftsquigarrow$ **normal**
$y \leftsquigarrow$ **normal**
$t = 5 \cdot x + 0.1 \cdot y$
factor $\langle t < -3 \rangle$
$t' = -3$
$p = (x, y)$

density

$t' \leftsquigarrow \mathrm{B}$
divide (**ret** -3) B t'
$x \leftsquigarrow$ **normal**
$y \leftsquigarrow$ **normal**
$t = 5 \cdot x + 0.1 \cdot y$
factor $\langle t < -3 \rangle$
$p = (x, y)$

$x \leftsquigarrow$ **normal**
$y \leftsquigarrow$ **normal**
$t = 5 \cdot x + 0.1 \cdot y$
factor $\langle |t| \leq 3 \rangle$
$p = (x, y)$

density

$x \leftsquigarrow$ **normal**
$y \leftsquigarrow$ **reparam** $(y \mapsto 5 \cdot x + 0.1 \cdot y)$ B
$t = 5 \cdot x + 0.1 \cdot y$
factor (10 / **jacobian** $(y \mapsto \cdots)$ B t)
divide lebesgue B t
factor (**dnorm** y)
factor $\langle |t| \leq 3 \rangle$
$p = (x, y)$

reparameterize

$t \leftsquigarrow \mathrm{B}$
$x \leftsquigarrow$ **normal**
$y = 10 \cdot (t - 5 \cdot x)$
factor 10
divide lebesgue B t
factor (**dnorm** y)
factor $\langle |t| \leq 3 \rangle$
$p = (x, y)$

Base-inferring disintegration

$x \leftsquigarrow$ **normal**
$y \leftsquigarrow$ **normal**
$t = 5 \cdot x + 0.1 \cdot y$
factor $\langle t < -3 \rangle$
$t' = -3$
$p = (x, y)$

density

$t' \leftsquigarrow \mathtt{B}$
divide (**ret** -3) $\mathtt{B}\ t'$
$x \leftsquigarrow$ **normal**
$y \leftsquigarrow$ **normal**
$t = 5 \cdot x + 0.1 \cdot y$
factor $\langle t < -3 \rangle$
$p = (x, y)$

$x \leftsquigarrow$ **normal**
$y \leftsquigarrow$ **normal**
$t = 5 \cdot x + 0.1 \cdot y$
factor $\langle |t| \leq 3 \rangle$
$p = (x, y)$

density

$x \leftsquigarrow$ **normal**
$y \leftsquigarrow$ **reparam** $(y \mapsto 5 \cdot x + 0.1 \cdot y)\ \mathtt{B}$
$t = 5 \cdot x + 0.1 \cdot y$
factor (10 / **jacobian** $(y \mapsto \cdots)\ \mathtt{B}\ t$)
divide lebesgue $\mathtt{B}\ t$
factor (**dnorm** y)
factor $\langle |t| \leq 3 \rangle$
$p = (x, y)$

reparameterize

$t \leftsquigarrow \mathtt{B}$
$x \leftsquigarrow$ **normal**
$y = 10 \cdot (t - 5 \cdot x)$
factor 10
divide lebesgue $\mathtt{B}\ t$
factor (**dnorm** y)
factor $\langle |t| \leq 3 \rangle$
$p = (x, y)$

Base-inferring disintegration

Infer principal base measure

$$\textbf{divide}\begin{pmatrix}\textbf{ret}\ -3 & \oplus\\ \textbf{ret}\ +3 & \oplus\\ \textbf{lebesgue} & \end{pmatrix}\ \mathtt{B} \quad \begin{cases}\textbf{divide}\ (\textbf{ret}\ -3)\ \mathtt{B}\ t' \\ \textbf{divide}\ (\textbf{ret}\ +3)\ \mathtt{B}\ t' \\ \textbf{divide lebesgue}\ \mathtt{B}\ t\end{cases}$$

Generalized density calculation

$$\begin{aligned}
&\mathit{density} : \mathbb{M}\,\alpha \to \mathbb{M}\,\alpha \to \alpha \to \{\mathbb{R}_+\} \\
&\mathit{density}\ \mu\ \nu\ t = \text{let}\ \mu' = \mu \otimes \mathbf{return}\ () \\
&\qquad\qquad\qquad\qquad\ \nu' = \nu \otimes \mathbf{return}\ () \\
&\qquad\qquad\qquad\qquad\ \lambda\ = \mathit{infer}\ \mu'\ t + \mathit{infer}\ \nu'\ t \\
&\qquad\qquad\qquad\quad \text{in}\ |\mathit{check}\ \mu'\ \lambda\ t| \div |\mathit{check}\ \nu'\ \lambda\ t|
\end{aligned}$$

Generalized MCMC sampling

$$
\begin{aligned}
&\mathit{mhg} : \mathbb{M}\,\alpha \to (\alpha \to \mathbb{M}\,\alpha) \to \alpha \to \mathbb{M}\,\alpha \\
&\mathit{mhg}\;\eta\;Q\;x = \mathbf{do}\;\{y \leftsquigarrow Q\;x; \\
&\qquad r_{xy} \leftsquigarrow \mathbf{return}\;(\mathit{greenRatio}\;\eta\;Q\;(x,y)); \\
&\qquad a_{xy} \leftsquigarrow \mathbf{return}\;\min(1, r_{xy}); \\
&\qquad b \leftsquigarrow \mathbf{bern}\;a_{xy}; \\
&\qquad \mathbf{return}\;(\mathbf{if}\;b\;\mathbf{then}\;y\;\mathbf{else}\;x)\}
\end{aligned}
$$

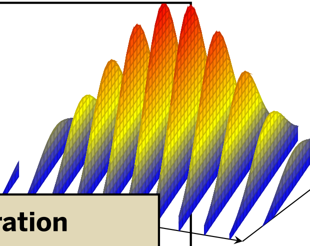
$$
\begin{aligned}
&\mathit{greenRatio} : \mathbb{M}\,\alpha \to (\alpha \to \mathbb{M}\,\alpha) \to (\alpha \times \alpha) \to \{\mathbb{R}_+\} \\
&\mathit{greenRatio}\;\eta\;Q = \mathit{density}\;(\mathit{fmap}\;\mathit{switch}\;(\eta \otimes\!\!= Q))\;(\eta \otimes\!\!= Q)
\end{aligned}
$$

$$
\begin{aligned}
&\mathit{switch} : (\alpha \times \beta) \to (\beta \times \alpha) \\
&\mathit{switch}\;p = (\mathrm{snd}\;p, \mathrm{fst}\;p)
\end{aligned}
$$

$$
\begin{aligned}
&(\otimes\!\!=) : \mathbb{M}\,\alpha \to (\alpha \to \mathbb{M}\,\beta) \to \mathbb{M}\,(\alpha \times \beta) \\
&m \otimes\!\!= k = \mathbf{do}\;\{x \leftsquigarrow m; y \leftsquigarrow k\;x; \mathbf{return}\;(x,y)\}
\end{aligned}
$$

Exact calculation

for approximate computation



https://github.com/pravnar/disintegrating-mixtures

Automatic differentiation

for gradient descent

Automatic disintegration

for inference and sampling

Generalizes density, conditioning

Derived by equational reasoning

A variety of base measures

Mixtures, disjoint sums,
dependent products

Infer principal base measure